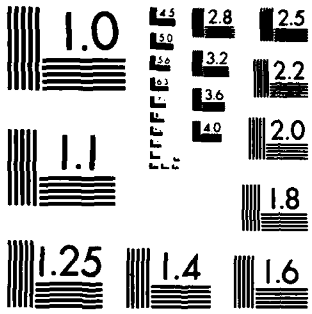
AD-A130 043 NUMERICAL SIMULATION OF SPACECRAFT CHARGING PHENOMENA AT HIGH ALTITUDE..(U) YORK UNIV DOWNSVIEW (ONTARIO) CENTRE FOR RESEARCH IN EXPERIME..

1/2

UNCLASSIFIED J G LAFRAMBOISE ET AL. 10 AUG 82 F/G 22/2 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

YORK UNIVERSITY

TENTANDA VIA

NUMERICAL SIMULATION OF SPACECRAFT CHARGING
PHENOMENA AT HIGH ALTITUDE

J.G. Laframboise, M. Kamitsuma, S.M.L. Prokopenko,
Jen-Shih Chang, and R. Godard

Physics Department and Centre for Research in Experimental Space Science
York University
Toronto, Canada M3J 1P3

10 August 1982

FINAL REPORT
1 March 1976 through 31 August 19[illegible]
Grant AFOSR-76-2962

Prepared for
Air Force Office of Scientific Research (AFSC)
Bolling Air Force Base, D.C., U.S.A. 20332

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER AFOSR-TR- 83-0549	2. GOVT ACCESSION NO. AD-A130043	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) NUMERICAL SIMULATION OF SPACECRAFT CHARGING PHENOMENA AT HIGH ALTITUDE		5. TYPE OF REPORT & PERIOD COVERED Final Report 1 March 1976 - 31 August 1981
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) J.G. Laframboise, M. Kamitsuma, S.M.L. Prokopenko, Jen-Shih Chang, and R. Godard		8. CONTRACT OR GRANT NUMBER(s) AFOSR-76-2962
9. PERFORMING ORGANIZATION NAME AND ADDRESS York University, Toronto, Canada M3J 1P3		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61102F 2311/A1
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Office of Scientific Research (AFSC) Bolling Air Force Base, D.C., U.S.A. 20332 Monitor/H. Radoski/NP		12. REPORT DATE 10 August 1982
		13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Numerical Simulation, Spacecraft Charging, Spacecraft - Environment Interactions

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes work done under grant AFOSR-76-2962. This work has included the development of computer programs for simulating spacecraft charging at three levels of complexity: LOCHG, a relatively simple local-charging calculation; CYLVIA, a two-dimensional simulation program for treating cylindrical spacecraft cross-sections, and XYCIC, a simulation program for the treatment of a larger variety of two-dimensional geometries. This work has also included studies of two physical phenomena which are fundamental to an improved understanding of spacecraft charging: the "threshold temperature" effect (over)

DD FORM 1 JAN 73 1473

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

and the "barrier" effect. Also included is a derivation of two results which appear likely to be of use in future simulation studies: an analytic expression for photoelectron currents on surfaces with variable illumination in electric fields, and a perturbation technique for calculating space-charge density and flux along particle orbits.

NUMERICAL SIMULATION OF SPACECRAFT CHARGING

PHENOMENA AT HIGH ALTITUDE

J.G. Laframboise, M. Kamitsuma, S.M.L. Prokopenko,

Jen-Shih Chang, and R. Godard

Physics Department and Centre for Research in Experimental Space Science

York University

Toronto, Canada M3J 1P3

10 August 1982

Final Report

1 March 1976 through 31 August 1981

Grant AFOSR-76-2962

Prepared for

Air Force Office of Scientific Research (AFSC)

Bolling Air Force Base, D.C., U.S.A. 20332

TABLE OF CONTENTS

ACKNOWLEDGMENTS

We are indebted to H.I. Cohen, L.W. Parker, and A.D. Stauffer for valuable discussions, and to M. Mandell for providing unpublished data used in Fig. 4.7 and discussed in Sec. 4.5. One of us (J.G. Laframboise) wishes to acknowledge the contributions of co-authors to specific sections of this Report, as follows: M. Kamitsuma (Secs. 3, 4, 5, and 6.3; Appendices B, D, and E), S.M.L. Prokopenko (Secs. 2 and 4; Appendices A, C, and D), Jen-Shih Chang (Sec. 2), and R. Godard (Secs. 2, 4, and 6.3; Appendices C and D). We wish to thank the computer center of Air Force Geophysics Laboratory for the provision of computer time, and the Department of Computer Services of York University for the use of its facilities. Special thanks are due to S. Bredesen of AFGL for much assistance with computing arrangements. We are grateful to the National Center for Atmospheric Research for the provision of computer software.

This work was supported by the U.S. Air Force Office of Scientific Research under grant number AFOSR-76-2962.

ABSTRACT

This report describes work done under grant AFOSR-76-2962. This work has included the development of computer programs for simulating spacecraft charging at three levels of complexity: LOCHG, a relatively simple local-charging calculation; CYLVIA, a two-dimensional simulation program for treating cylindrical spacecraft cross-sections, and XYCIC, a simulation program for the treatment of a larger variety of two-dimensional geometries. This work has also included studies of two physical phenomena which are fundamental to an improved understanding of spacecraft charging: the "threshold temperature" effect and the "barrier" effect. Also included is a derivation of two results which appear likely to be of use in future simulation studies: an analytic expression for photoelectron currents on surfaces with variable illumination in electric fields, and a perturbation technique for calculating space-charge density and flux along particle orbits.

1. INTRODUCTION

The performance of many satellites in geostationary orbit has been degraded by anomalous events which include frequent spurious spacecraft commands and in some cases permanent damage [Rosen, 1975; McPherson and Schober, 1976]. These events invariably appear to involve electrical discharges caused by differential charging of spacecraft surfaces to large relative potentials. The latter condition is known to result from the large average energies (up to a few keV) of the charged-particle environment at geostationary-orbit altitude, particularly in disturbed magnetospheric conditions. In order to be able to design spacecraft in future which do not suffer from such difficulties, it is important to be able to make reliable predictions of them on proposed spacecraft configurations. The ability to make such predictions requires improved understanding of the plasma sheath which connects a spacecraft with its environment in such conditions. This sheath is more complex in many ways than those which usually surround lower-altitude spacecraft because at geostationary-orbit altitude, the fluxes of incident electrons, secondary and backscattered electrons, and photoelectrons can all have comparable magnitudes and usually vary by large amounts over the surface of a spacecraft, and the relative weakness of space-charge shielding (large Debye length) means that electric fields due to charging of one part of a spacecraft surface can readily extend around the spacecraft to other parts, including those on its other sides, exerting strong influences on charged-particle collection by them and hence on their charging.

Because of the magnitude of the spacecraft-charging problem, a joint USAF-NASA programme was commenced during the 1970-1980 decade to study it, including the launching of the P78-2 (SCATHA) satellite, which was specifically devoted to the study of the high-altitude charging process, and also including theoretical work on the nature of the spacecraft-environment interface problem: this work was to be closely coordinated with analysis of SCATHA observations.

The work described in this report is part of this theoretical effort.

This work has been conducted using a number of different approaches. In Sec. 2, we describe a relatively simple local-current-balance calculation of charging which formed the first phase of it, and led to the discovery that floating potentials (voltages) of many spacecraft surface materials can have more than one possible value in certain, frequently-occurring, space environments; the possibility that this might occur had been earlier predicted by Whipple (1965). Attempts to gain further understanding of the conditions in which this might occur led to the realization that this was one of several closely-related phenomena which could all be explained in terms of a newly-defined property of spacecraft surface materials, the threshold material temperature for high-voltage charging (Sec. 3). We present herein a table of these threshold temperatures, calculated for a variety of spacecraft surface materials. A second phase of our effort was the development of a more elaborate, two-dimensional spacecraft charging simulation program called CYLVIA, which treats the regions around circular spacecraft cross-sections (Sec. 4). We have also begun development of another two-dimensional simulation program, called XYCIC (Sec. 5), which treats a wider class of two-dimensional geometries than does CYLVIA. The effort to develop CYLVIA, together with efforts by other workers (Sec. 4.6), led gradually to an understanding of the importance of the barrier effect, an example of which is presented in Sec. 4.6.

A related phase of our work has been an effort to develop efficient numerical techniques, and analytical replacements for some numerical procedures, which will be of general use not only in our own simulation programs but also in other spacecraft-charging simulations. To this end, we have developed an analytic expression for the electric current produced by photoelectron migration on a surface with spatially-varying illumination in an electric field (Sec. 6), and a perturbation technique for calculating current

density and space-charge density carried by collisionless charged-particle orbits (Sec. 7).

Listings of computer programs developed as part of this work appear in Appendices A-E.

2. LOCHG: A LOCAL-CHARGING CALCULATION

2.1 INTRODUCTION

In order for high-voltage differential charging of a spacecraft to occur, some of its surfaces must charge to large (usually negative) potentials (voltages) relative to space. The amount of such "absolute" charging, for the surface at the largest potential, is generally simpler to estimate than are the differential potentials between this surface and others. On the sunlit side of a spacecraft, photoelectron emission tends to compensate for incident electron fluxes. Sunlit-side surface potentials therefore are generally less negative than shaded-side potentials, and may be slightly positive if photoemission flux > ambient random electron flux. Therefore, an approximate upper bound on differential charging magnitudes can be obtained by simply calculating floating potentials of electrically-isolated shaded surfaces, relative to space potential. In the present work, we have attempted to obtain upper bounds on these floating potentials, which in cases of interest are usually highly negative, because these bounds constitute "worst cases" for design purposes, and also because unlike more exact calculations, they can be obtained from simple current-balance calculations. Furthermore, it is sufficient to consider local current balance only, because this corresponds to an electrically-isolated surface element, which is also a "worst case" for differential charging. To calculate these bounds, we have constructed a computer program called LOCHG (LOcal CHarGing), which extends a previous calculation by Knott [1972], of the floating potential of a spherically-symmetric geostationary-altitude satellite in eclipse. A listing of LOCHG appears in Appendix A. To investigate geometrical effects, we have replaced Knott's use of the Mott-Smith and Langmuir [1926] orbit-limited current expression for collection of Maxwellian ions by a unipotential sphere, by the corresponding expression for an infinite cylinder;

both expressions have been shown [Laframboise and Parker, 1973] to be upper bounds for collisionless ion collection as a function of local surface potential, for three- and two-dimensional collectors, respectively, regardless of collector shape, sheath potential, or potential of other parts of the collector. This replacement causes a large decrease in ion collection and a correspondingly large increase in negative shaded-side floating potentials (Sec. 2.3). Another important ion-current restriction may be caused by "effective-potential barrier" or "angular-momentum selection" effects [Bernstein and Rabinowitz, 1959; Laframboise, 1966; Laframboise and Parker, 1973], in which the presence of less-negative sunlit-side potentials produces dipole and higher moments in the sheath potential [Fahleson, 1973], causing steepening and contraction of the potential well surrounding the shaded side (Fig. 2.1). A similar steepening effect will also occur if an isolated shaded surface element is surrounded by adjacent shaded surfaces which for any reason have less-negative potentials (Fig. 2.2). The most extreme possibility would be a potential profile which was equal to space potential almost to the spacecraft surface, then fell discontinuously to surface potential. This limit would correspond to a "planar sheath" situation in which the ion collection on any shaded convex surface would be given by just the ion random flux. This amounts to a further ion-current restriction which produces even larger increases in negative shaded-side floating potentials (Sec. 2.3). This situation corresponds to a velocity-space cutoff boundary for incident ions which is "one-dimensional"; the cutoff boundaries corresponding to spherical and infinite cylindrical collectors are, respectively, "three-dimensional" and "two-dimensional" (Sec. 2.2) [Laframboise and Parker, 1973].

We also show (Sec. 2.3) that if shaded cavities containing isolated surfaces exist on a spacecraft, negative potentials on such surfaces may surpass even these predictions. In some cases, more than one possible floating potential results from the calculation. This has several implications (Sec. 2.3), including the possibility

(a) of "bifurcation phenomena" in which adjacent isolated surfaces of the same material may arrive at different floating potentials as a result of differences in their charging histories (b) that large and relatively sudden changes in surface potentials may result from gradual changes in ambient velocity distributions (c) that such changes may also result from relatively gradual changes in beam emission currents in a beam experiment. In Sec. 2.4 we calculate effects of ion drift motion on floating potentials.

We have also modified Knott's calculation in another way, by including currents due to electron backscattering (Sec. 2.2). These currents will tend to decrease net electron collection, thereby making floating potentials less negative than otherwise (Sec. 2.3). A process not included by either Knott or ourselves is secondary electron emission due to ion impacts; this will also tend to make floating potentials less negative. Ion-produced secondaries have been included in a calculation by DeForest [1972] of the floating potential of a shaded aluminum surface. However, no direct comparison is possible between his result and ours (Sec. 2.3) because we have used different ambient velocity distributions than his.

2.2 THEORY

The ambient electron energy distributions used in the present work are a model quiet-time spectrum (Knott [1972], Figure 1) and a model disturbed spectrum (Knott [1972], Figure 2b) based on measurements by Shield and Frank [1970] and DeForest and McIlwain [1971], respectively. Both of these distributions, and also the ambient ion distribution, are assumed isotropic. The disturbed spectrum was chosen from the three used by Knott because it has a higher average electron energy (∿4keV) than the others. In using it, we have changed it as follows: in the energy ranges $0.5 \text{ keV} \leq E \leq 10$ keV and $10 \text{ keV} \leq E \leq 40$ keV, we have replaced Knott's differential energy spectrum by $\sqrt{2} \times 10^8 E^{-1/2}$ and $\sqrt{2} \times 10^9 E^{-3/2}$ electrons/cm^2 sec steradian keV, respectively, where E is

energy. These relations are simpler than those indicated by Knott, and they also bring the model spectrum into closer agreement with the data on which it is based. We therefore believe that they may have been the ones actually used by Knott, and that the corresponding parts of Figure 2b in his paper may be incorrectly plotted. For any spacecraft surface having a negative potential $\phi_s < 0$, or for a three-dimensional (e.g. spherical) surface having $\phi_s > 0$, the orbit-limited flux (particle current density) J_e of ambient electrons is given by [Laframboise and Parker, 1973]:

$$
\begin{aligned}
J_e &= \int f v_n \, d^3 \vec{v} \\
&= \int_{E=\max(0,-e\phi_s)}^{E=\infty} \int_{\theta=0}^{\theta=\pi/2} \int_{\psi=0}^{\psi=2\pi} f(E)(v \cos\theta)(v^2 \sin\theta \, dv \, d\theta \, d\psi) \\
&= \int_{\max(0,-e\phi_s)}^{\infty} (1 + e\phi_s/E)(dJ_{eo}/dE) \, dE \qquad (2.1)
\end{aligned}
$$

where e is magnitude of unit electronic charge, ϕ_s is local surface potential, dJ_{eo}/dE is the ambient energy-differential flux incident on one side of an arbitrarily oriented surface element, v_n is the inward velocity component normal to the same surface element, (v,θ,ψ) are spherical coordinates in velocity space with v_n as polar axis, and $E = \frac{1}{2}m_e v^2 - e\phi$. dJ_{eo}/dE is π times the energy-differential flux per steradian used by Knott [1972], and is given in terms of the ambient electron velocity distribution $f = d^3N_\infty/d^3\vec{v}$ by the relation $dJ_{eo}/dE = 2\pi f E/m_e^2$, where m_e is electron mass and N_∞ is ambient ion or electron number density. Since f is isotropic, $f \equiv f(E)$. The factor $(1 + e\phi_s/E)$ in (2.1), which appears to have been neglected by Knott, leads to a divergent integration in (2.1) if $\phi_s > 0$, unless $dJ_{eo}/dE \to 0$ as $E \to 0$, i.e. f(E) remains finite as $E \to 0$. This implies that the differential fluxes in Knott's Figures 1 - 3 must approach zero linearly with E at E values smaller than those shown in these Figures. In the present work we have introduced a linear rise in dJ_{eo}/dE from 0 to 1 eV. We have also introduced a sharp upper cutoff at 50 keV in the quiet-time spectrum, also in order to avoid a divergent integration when calculating average energy for use in backscattering calculations (see below). The resulting values of N_∞ are 5.43 cm^{-3} and 5.39 cm^{-3} for the quiet-time

and disturbed spectra, respectively. These values are consistent with those quoted by Knott [1972] for the corresponding spectra. The resulting values of the average ambient electron energy $\overline{E}$ are 0.176 keV and 4.17 keV, respectively. The incorporation of these changes has a relatively minor influence on Knott's results [Prokopenko and Laframboise, 1977, Table 1].

In order to obtain the orbit-limited electron flux expression for an arbitrary cylindrical collector, the lower integration limit in (2.1) must be replaced by the two-dimensional velocity-space cutoff boundary $E_\perp = \max(0, -e\phi_s)$, where $E_\perp$ is the total energy of transverse motion $\frac{1}{2}m_e(v_x^2 + v_y^2) - e\phi$, and we have chosen a z coordinate perpendicular to the cylinder cross-section. If $\phi_s > 0$, this complicates the integration in (2.1), which may then be done in either of two ways. The first [Laframboise and Parker, 1973, Eq. (6)] is to convert (2.1) into an integration using cylindrical coordinates in velocity space. A more convenient method [Mott-Smith and Langmuir, 1926; Polychronopulos, 1973] is as follows. We choose rectangular coordinates (v_n, v_t, v_z) in velocity space, such that v_n is the velocity component in the inward normal direction at the collector surface. Then v_t and v_z become tangential coordinates, with v_t in the plane of the cylinder cross-section. We then transform to spherical coordinates (v,θ,ψ) with v_z as polar axis. Then: $v_z = v\cos\theta$, $v_n = v\sin\theta\cos\psi$, and $v_t = v\sin\theta\sin\psi$. The condition $E_\perp \geq 0$ is equivalent to $\sin\theta \geq [e\phi_s/(E + e\phi_s)]^{\frac{1}{2}}$. For $\phi_s > 0$, Eq. (2.1) is then replaced by:

$$J_e = \int f v_n \, d^3\vec{v}$$

$$= 2\int_{E=0}^{E=\infty} \int_{\psi=-\pi/2}^{\psi=\pi/2} \int_{\theta=\mathrm{Arc}\,\sin[e\phi_s/(E+e\phi_s)]^{\frac{1}{2}}}^{\theta=\pi/2} f(E)(v\sin\theta\cos\psi)(v^2\sin\theta\, dv\, d\theta\, d\psi)$$

$$= \int_0^\infty \frac{2}{\pi}\left[\mathrm{Arc}\,\sin\left(\frac{E}{E+e\phi_s}\right)^{\frac{1}{2}} + \frac{(Ee\phi_s)^{\frac{1}{2}}}{E+e\phi_s}\right]\left(1 + \frac{e\phi_s}{E}\right)\frac{dJ_{eo}}{dE}\, dE \quad . \qquad (2.2)$$

In comparison with Eq. (2.1), we see that the integrand in (2.2) contains an extra, energy-dependent weighting factor, which arises from integration of v_n over the fractional solid angle over which ambient electrons can reach the collector at each energy.

A similar procedure is advantageous in obtaining the one-dimensional (Sec. 2.1) orbit-limited flux expression. In this case, the lower limit in (2.1) must be replaced by: $E_n = \max(0, -e\phi_s)$, where $E_n = \frac{1}{2} m_e v_n^2 - e\phi$. This time we transform (v_n, v_t, v_z) to spherical coordinates (v, θ, ψ) with v_n as polar axis. The condition $E_n \geq 0$ is equivalent to $\cos\theta \geq [e\phi_s/(E + e\phi_s)]^{\frac{1}{2}}$. For $\phi_s > 0$, Eq. (2.1) is now replaced by:

$$J_e = \int_{E=0}^{E=\infty} \int_{\theta=0}^{\theta=\mathrm{Arc}\cos[e\phi_s/(E+e\phi_s)]^{\frac{1}{2}}} f(E)(v\cos\theta)(2\pi v^2 \sin\theta \, dv \, d\theta)$$

$$= \int_0^{\infty} \frac{dJeo}{dE} \, dE \tag{2.3}$$

independently of collector potential, as expected.

The corresponding expressions for ion flux J_i are simpler because the ions are assumed to be Maxwellian. Corresponding to the three-, two-, and one-dimensional velocity-space cutoffs described above, we obtain, respectively [Mott-Smith and Langmuir, 1926; Laframboise and Parker, 1973], for ion-attracting surface potentials $\chi_{is} > 0$:

$$J_i = J_{io} \begin{cases} (1 + \chi_{is}) & (2.4) \\ [2(\chi_{is}/\pi)^{\frac{1}{2}} + \exp(\chi_{is}) \operatorname{erfc}(\chi_{is}^{\frac{1}{2}})] & (2.5) \\ (1) & (2.6) \end{cases}$$

where $\chi_{is} = -e\phi_s/kT_i$, k is Boltzmann's constant, T_i is ion temperature and J_{io} is the ion random flux $N_\infty(kT_i/2\pi m_i)^{\frac{1}{2}}$. For ion-retarding surface potentials $\chi_{is} < 0$, we obtain:

$$J_i = J_{io} \exp(\chi_{is}). \tag{2.7}$$

We have assumed, following Knott [1972], that $T_i = 1$ keV, and that the random ion-to-electron flux ratio $J_{io}/J_{eo} = 0.025$. Making these two assumptions simultaneously causes ambient charge neutrality to be violated in general. Large apparent violations of charge neutrality are frequently observed in measurements made by particle energy analyzers. Such discrepancies are believed to result from failure to measure particles outside the energy ranges of these analyzers, especially at energies of a few volts or less [DeForest and McIlwain, 1971]. Our calculations therefore neglect any current contributions which may be made by such particles.

For the secondary electron fractional yield $\delta(E)$, we have used, following Knott [1972], the relation of Sternglass [1954a]:

$$\delta(E) = 7.4\ \delta_{max}\ (E/E_{max})\ \exp\left[-2(E/E_{max})^{1/2}\right]. \tag{2.8}$$

We have used values of δ_{max} and E_{max} from Gibbons [1966], Hachenberg and Brauer [1959], and Willis and Skinner [1973].

The process of electron backscattering, which was not included in Knott's calculations, becomes important at incident electron kinetic energies larger than those for which secondary emission is dominant. For the backscattered electron fractional yield η, we have fitted the results of Sternglass [1954b] and Palluel [1947] with a relation of the form:

$$\eta(E) = A - B\exp(-CE) \tag{2.9}$$

where the coefficients A, B, and C are functions of the atomic number Z of the surface material. We have evaluated A, B and C for each surface material considered (Sec. 2.3) by substituting Sternglass' and Palluel's values of η at 0, 1 and 16 keV, into Eq. (2.9). For compound surface materials, we have used an atomic number given by a weighted average of those of each constituent. There exist more recent measurements of η [Thomas and Pattinson, 1970; Darlington and Cosslett, 1972] which give generally larger values than those of Sternglass or Palluel, especially for electrons having near-tangential incidence.

Increased values of δ have also been measured for electrons having non-normal incidence [Allen, 1939; Dekker, 1958]. In a separate calculation (Sec. 3), we have included angle-of-incidence effects on η and δ, but here we have ignored these. We have therefore underestimated both δ and η, and our predicted floating potentials in Sec. 2.3 will therefore be somewhat more negative than more realistic corresponding values.

When $\phi_s > 0$, not all secondary and backscattered electrons will escape. To calculate flux escaping, we assume [Sternglass, 1954b; Chung and Everhart, 1974] that both secondary and backscattered electrons are emitted with Maxwellian velocity distributions having thermal energies $E_{sec} = kT_{sec} = 3$ eV, and $E_{scat} = kT_{scat} = (0.45 + 2 \times 10^{-3}Z)\overline{(E + e\phi_s)}$ eV, respectively, regardless of the form of the incident velocity distribution. Here, $\overline{(E + e\phi_s)}$ is the average incident electron kinetic energy. We further assume that escape of emitted electrons is orbit-limited, i.e. that no barriers of effective potential [Bernstein and Rabinowitz, 1959; Laframboise, 1966; Laframboise and Parker, 1973] or negative barriers of electric potential exist on the shaded side. {Fahleson [1973] has pointed out that such a barrier is likely to exist on the sunlit side independently of any space-charge effects, if substantial shaded-sunlit differences exist in ϕ_s. Such a barrier would cause most electrons emitted from the sunlit side to be recollected, driving sunlit-side potential just negative enough to almost destroy the barrier [Sec. 4.5]; [Katz et al, 1979, Figs. 17-21]}. The expressions for the escaping secondary and backscattered fluxes J_{sec} and J_{scat} therefore are:

$$J_{sec} = \int_{-e\phi_s}^{\infty} \delta(E + e\phi_s)(1 + e\phi_s/E)(dJ_{eo}/dE)\, dE \tag{2.10}$$

$$J_{scat} = \int_{-e\phi_s}^{\infty} \eta(E + e\phi_s)(1 + e\phi_s/E)(dJ_{eo}/dE)\, dE \tag{2.11}$$

if $\phi_s < 0$. If $\phi_s > 0$, the three-, two-, and one-dimensional cases must be considered separately. We define $\chi_{sec} = e\phi_s/kT_{sec}$ and $\chi_{scat} = e\phi_s/kT_{scat}$. For brevity, we consider only the secondary fluxes; the corresponding results for backscattered fluxes may be obtained by replacing δ by η and χ_{sec} by χ_{scat} throughout. If J_s is the emitted flux of secondaries,

then their velocity distribution at the surface is $f_s = (J_s/2\pi)(m_e/kT_{sec})^2$ $\exp(-\frac{1}{2}m_e v^2/kT_{sec})$. In the three-dimensional case, the cutoff condition for their escape is $E = \frac{1}{2}m_e v^2 - e\phi_s > 0$. We redefine v_n as velocity component in the outward normal direction, and we use spherical coordinates as defined in connection with (2.3). We obtain, for the escaping secondary flux:

$$J_{sec} = \int f_s \, v_n \, d^3\vec{v}$$

$$= \frac{J_S}{2\pi}\left(\frac{m_e}{kT_{sec}}\right)^2 \int_{E=0}^{E=\infty} \int_{\theta=0}^{\theta=\frac{\pi}{2}} \exp(-\tfrac{1}{2}m_e v^2/kT_{sec})(v\cos\theta)(2\pi v^2 \sin\theta \, dv \, d\theta)$$

$$= (1+\chi_{sec})\exp(-\chi_{sec}) \int_0^\infty \delta(E+e\phi_s)(1+e\phi_s/E)(dJ_{eo}/dE)dE. \qquad (2.12)$$

The factor $(1+\chi_{sec})$ is noteworthy, because it is specific to three-dimensional, as opposed to planar, sheath geometry. In the two-dimensional case, the cutoff condition for escape is $\frac{1}{2}m_e(v_n{}^2+v_t{}^2)-e\phi_s > 0$, and the integral for J_S contains the extra weighting factor which appears in Eq. (2.2). It is convenient to use spherical coordinates as defined in connection with (2.2). We obtain:

$$J_{sec} = \left[2(\chi_{sec}/\pi)^{\frac{1}{2}} + \exp(\chi_{sec})\,\mathrm{erfc}(\chi_{sec}{}^{\frac{1}{2}})\right]\exp(-\chi_{sec})$$

$$\times \int_0^\infty \frac{2}{\pi}\left[\mathrm{Arc\ sin}\left(\frac{E}{E+e\phi_s}\right)^{\frac{1}{2}} + \frac{(Ee\phi_s)^{\frac{1}{2}}}{E+e\phi_s}\right](1+\frac{e\phi_s}{E})\ \delta(E+e\phi_s)\frac{dJ_{eo}}{dE}\,dE\ . \qquad (2.13)$$

In the one-dimensional case, the escape condition is $\frac{1}{2}m_e v_n{}^2 - e\phi_s > 0$, and we again use spherical coordinates as defined in connection with (2.3). We obtain:

$$J_{sec} = \exp(-\chi_{sec}) \int_0^\infty \delta(E+e\phi_s)(dJ_{eo}/dE)\ dE\ . \qquad (2.14)$$

The floating potential(s) of an isolated shaded surface element is (are) now given by the zero(s) of the function:

$$J_{net} = J_i - J_e + J_{sec} + J_{scat}. \qquad (2.15)$$

2.3 RESULTS AND DISCUSSION: SHADED CAVITIES AND MULTIPLE FLOATING POTENTIALS

Table 1 shows floating potential values obtained using the program LOCHG, which performs numerical solution of the equation $J_{net} = 0$, where J_{net} is given by Eq. (2.15) An important feature of Table 1 is the very

large floating potentials which are evident in disturbed conditions in the presence of the two- and one-dimensional velocity-space cutoffs. The dramatic differences among these results are evidence that spacecraft geometry and sheath potential shape are important influences in determining floating potentials. As floating potential becomes more negative, it also becomes more sensitive to the presence of small amounts of high-energy electrons. This means that if a spacecraft should encounter conditions that are "more disturbed" than those given by Knott's spectrum 2b, the values in Table 1 most likely to be significantly exceeded are those for the one-dimensional cutoff. This implies that for design purposes in which worst-case information is desired, it is important to do calculations with the "most disturbed" electron spectra available. In obtaining these results, we have made no attempt to calculate the charging times involved.

Also evident in Table 1 are situations in which the current-voltage characteristic of the surface has three roots. For these to occur, it is necessary that δ_{max} be substantially greater than one, and that the ambient electron spectrum be at least slightly non-Maxwellian (Sec. 3). The latter requirement arises because if the incident electrons are Maxwellian and ion-produced secondaries are ignored, the ratio of total secondary emission current to incident electron current will then be independent of ϕ_s for $\phi_s < 0$, and the total secondary emission will therefore be a monotonic function of ϕ_s for $\phi_s < 0$. (An exception to this, in which a triple-root situation occurs with Maxwellian ambient electrons, has been found by Meyer-Vernet (1982), but this exception applies only when $T_e < T_{sec}$, a condition which is not applicable here; see also Sec. 3). The centre root never represents a possible floating potential, because it is "unstable" in the sense that a small change in surface potential would cause a net current collection of a sign which would drive the surface potential away from this root to one on either side. Various consequences of such a situation are discussed later in this Section. A similar phenomenon in electronic image storage devices has been discussed by Kazan and Knoll [1968, pp. 17-19].

In comparison with the results of Knott [1972], Table 1 includes the further addition of backscattered electron flux (Sec. 2.2). In most cases, the effect is a moderate reduction of negative floating potentials. In some cases, the reduction is large, as in the case of a gold surface

exposed to the "quiet" spectrum. In several other cases, all associated with the quiet spectrum, backscattering changes a multiple-root to a single-root situation [Prokopenko and Laframboise, 1977]. As indicated in Sec. 2.2, we have probably underestimated secondary and backscattered fluxes caused by electron impacts, and we have also (Sec. 2.1) ignored secondary electron emission caused by ion impacts. Both of these effects would tend to further reduce negative potentials. For ion-produced secondaries, fractional yields at energies from 5 to 50 keV are generally of order unity to several times unity for metallic surfaces [Cousinie et al, 1959; Ray and Barnett, 1971; Baragiola et al, 1979] so substantial reductions due to this cause can be expected whenever ion collection plays an important role in total current balance. In comparision with the results of Table 1, potentials reaching -19kV on the ATS 6 spacecraft in eclipse have been observed (E.C. Whipple, Jr., private communication).

Figures 2.3 - 2.7 show current-voltage characteristics for some of the situations in Table 1. Figure 2.3 shows a "typical" single-root situation in which secondary and backscatter contributions do not change the general shape of the net current curve. Figure 2.4 shows the above-mentioned case of gold exposed to the quiet spectrum, in which the backscatter contribution changes a large predicted negative floating potential to a much smaller value. Figure 2.5 shows a triple-root situation. Figure 2.6 shows the disappearance of a triple-root situation because of backscatter. In Fig. 2.7, secondary electron current is sufficient by itself to prohibit a negative floating potential.

We now examine situations which may arise in the case of spacecraft which have shaded cavities containing electrically-isolated interior surfaces. Figure 2.8 shows an idealization of such a spacecraft. We wish to show that the effects of surface concavity may cause ion collection to be reduced more than net electron collection at an interior point such as B, relative to an exterior point A; such a situation would result in floating potentials more negative than those of Table 1. To demonstrate this possibility, we first note that in the presence of an isotropic ambient plasma, incident fluxes to any surface depend only [Laframboise and Parker, 1973] on the locations, in velocity space, of

the cutoff boundaries inside of which the orbits of ambient particles can connect "from infinity" to the surface. Figure 2.8 shows a set of the associated "cutoff orbits". We see from Fig. 2.8 that the included angle between cutoff orbits has been reduced in going from A to B for ions but not for electrons, for which orbits tangential to the surface are shown as reaching both A and B and the range of allowable directions remains 180°. Accordingly, the incident ion current contribution for the energy shown will also be reduced, but the electron contribution will not. This picture is invalid for higher-energy electrons at B, whose orbits are straighter and will have a greater tendency to connect back to the interior surfaces of the cavity. Even though such higher-energy orbits will generally have lower populations than lower-energy orbits, it is not clear whether the relative current reduction at B will be greater for ions or for electrons. Therefore, this argument demonstrates only the possibility that the bounds in Table 1 will be exceeded. On the other hand, this possibility will be enhanced by the effects of secondary and backscattered electrons, which will tend to be recollected inside any cavity, rather than escaping into space, thus tending to increase net electron collection and driving floating potentials more negative. This effect will be strongest for backscattered electrons because their higher emission energies will cause them to have straighter orbits. The detailed numerical simulation required to draw firm conclusions remains to be done. An additional feature of cavities is their generally higher outgassing pressures, which will increase any tendencies for arcing to occur. More negative floating potentials may also result if the ambient electron distribution contains beam-like constituents [DeForest, 1977] which happen to be directed into a cavity. Severe arcing problems are known to have occurred between electronic components mounted inside a cavity at one end of the DSCS spacecraft.

Finally, we discuss some further implications of the multiple-root results shown in Table 1 and Fig. 2.5. We first consider a situation involving a shaded, isolated spacecraft surface whose external conditions change with time, as in the case of time-varying ambient distributions,

in such a way as to produce an evolution from a multiple-root to a single-root situation. If a spacecraft surface were floating at a potential corresponding to a root which disappeared, a large and relatively sudden change in surface potential would occur, even if external changes were gradual. In many situations, rapid redistribution of potentials on other parts of the spacecraft would result. In order to examine this situation in more detail, we illustrate it schematically in Fig. 2.9. In Fig. 2.9a, the current-voltage curve at an earlier time t_1 has three roots. The two roots furthest apart, ϕ_{1a} and ϕ_{1b}, correspond to stable floating surface potentials; these are typically from one to ten kilovolts apart. As a result of a changing environment, this curve evolves continuously into the one shown for the later time t_2, which has only the single root ϕ_2. If the surface happened to be floating at the voltage ϕ_{1a} at time t_1, then a sudden positive change in surface voltage, toward the voltage ϕ_2, would occur as soon as the two left-hand roots disappeared.

In Fig. 2.9b, a spacecraft surface floating at the more positive root (at ϕ_{1b}) would undergo a sudden negative change in voltage, toward ϕ_2, as soon as the two right-hand roots disappeared. Sudden voltage changes have been observed both on the SCATHA (P78-2) satellite and in numerical simulations (Sec. 3.1; Schnuelle et al, 1981; Stannard et al, 1982).

Similar sudden changes could occur on a spacecraft which was emitting a changing ion or electron gun current, if an increase or decrease in gun current were to provide enough net current to a surface to cause it to change from one stable floating potential to another. Again, such a change would be relatively sudden even if beam current changes were gradual.

A different effect could arise if the situation evolved in reverse fashion, so that t_2 were now the earlier time. Consider a spacecraft whose rotation carries a number of independently-floating surface panels from sunlight into shade, one after another. Suppose the situation evolves as in Fig. 2.9b. A surface already in the shade when the environment changed would go from voltage ϕ_2 to ϕ_{1a}. However, a surface which was

still in the sun, and whose photoemission current held its voltage close to ϕ_{1b}, would go to ϕ_{1b} when it entered the shade. As a result, two adjacent surfaces made of the same material could easily come to quite different voltages in the same environment, with a resulting danger of arcing between them.

In Sec. 3, we examine in detail the conditions in which multiple roots can occur.

2.4 PREDICTION OF ION DRIFT EFFECTS ON SPACECRAFT FLOATING POTENTIALS

2.4.1 INTRODUCTION

The plasma environment of high-altitude spacecraft has been observed to involve ion velocity anisotropies which sometimes become comparable to ion mean thermal speeds. These anisotropies are varied in nature, but a qualitative estimate of their effects on spacecraft charging may be obtained by considering the effects of a relatively simple anisotropy, namely the addition of a drift to a Maxwellian velocity distribution. Such drifts may cause an electrically-isolated spacecraft surface to float at a substantially increased negative potential if it is simultaneously shaded and downstream relative to the drift direction. In this Section, we present a calculation of upper and lower bounds on such potentials for a spherical spacecraft, based on the fact that ion collection on the spacecraft at its downstream point is bounded above by the corresponding current which would be collected if the spacecraft were an equipotential (i.e. were more attractive for ions elsewhere on its surface than it is in reality) and bounded below by the corresponding result for a sphere at space potential. The results show that (1) the ion speed ratio at which drift effects become "important" (i.e. change the floating potential by at least 10%) can be as low as 0.1, and may be decreased if the ambient electrons are non-Maxwellian; (2) the effects of ion speed ratio increase with increasing ion-to-electron temperature ratio; (3) negative floating potentials for drifting Maxwellian ion velocity distributions with speed ratio unity are typically about twice as large as the corresponding potentials for nondrifting conditions. We now examine the effects of ion drift in detail.

If a spacecraft is exposed to ambient ions whose drift velocity U is comparable to or larger than their most probable thermal speed [ion speed ratio $S_i = U/(2kT_i/m_i)^{\frac{1}{2}} \gtrsim 1$, where k is Boltzmann's constant and m_i and T_i are ion mass and assumed ion temperature], a large decrease in ion flux J_i to downsteam surfaces will occur. Unless such surfaces are able to expel surplus incident electron fluxes, e.g. by photoemission,

2.4 PREDICTION OF ION DRIFT EFFECTS ON SPACECRAFT FLOATING POTENTIALS

2.4.1 INTRODUCTION

The plasma environment of high-altitude spacecraft has been observed to involve ion velocity anisotropies which sometimes become comparable to ion mean thermal speeds. These anisotropies are varied in nature, but a qualitative estimate of their effects on spacecraft charging may be obtained by considering the effects of a relatively simple anisotropy, namely the addition of a drift to a Maxwellian velocity distribution. Such drifts may cause an electrically-isolated spacecraft surface to float at a substantially increased negative potential if it is simultaneously shaded and downstream relative to the drift direction. In this Section, we present a calculation of upper and lower bounds on such potentials for a spherical spacecraft, based on the fact that ion collection on the spacecraft at its downstream point is bounded above by the corresponding current which would be collected if the spacecraft were an equipotential (i.e. were more attractive for ions elsewhere on its surface than it is in reality) and bounded below by the corresponding result for a sphere at space potential. The results show that (1) the ion speed ratio at which drift effects become "important" (i.e. change the floating potential by at least 10%) can be as low as 0.1, and may be decreased if the ambient electrons are non-Maxwellian; (2) the effects of ion speed ratio increase with increasing ion-to-electron temperature ratio; (3) negative floating potentials for drifting Maxwellian ion velocity distributions with speed ratio unity are typically about twice as large as the corresponding potentials for nondrifting conditions. We now examine the effects of ion drift in detail.

If a spacecraft is exposed to ambient ions whose drift velocity U is comparable to or larger than their most probable thermal speed [ion speed ratio $S_i = U/(2kT_i/m_i)^{1/2} \gtrsim 1$, where k is Boltzmann's constant and m_i and T_i are ion mass and assumed ion temperature], a large decrease in ion flux J_i to downsteam surfaces will occur. Unless such surfaces are able to expel surplus incident electron fluxes, e.g. by photoemission,

their floating potentials will become substantially more negative as a result. If the ambient electron temperature T_e is simultaneously large, or more generally the ambient electron energy distribution has a significant high-energy component, then large absolute increases in negative floating potentials will occur, with correspondingly increased arcing hazards. Even if T_e is relatively small, such effects may influence surface potentials enough to disturb particle and field measurements. S_i values of order unity may be reached in the Earth's outer magnetosphere (Mauk 1975; DeForest 1977, Figs. 6 and 8); larger values are likely in the outer Jovian magnetosphere and magnetosheath (Goldstein and Divine 1977), and in the solar wind (Dessler 1967; Axford 1968; Manka 1973). In both outer magnetospheres, electron distributions having substantial high-energy components have been observed (DeForest and McIlwain 1971; Goldstein and Divine 1977).

A calculation of ion drift effects on the floating potential of the lunar surface has been done by Manka (1973), using a local-current-balance formulation. Parker (1978) has done exact numerical calculations of floating surface potentials for nonconductive finite cylindrical objects, including photoemission due to illumination of one end and ion drift parallel to the axis of symmetry.

In this Section we have done an approximate calculation of ion drift effects on the floating potential of a shaded, downstream, electrically-isolated surface element on a spherical spacecraft (Fig. 2.10), using a local-current-balance formulation which yields upper and lower bounds on such potentials. This formulation is an adaptation of that of Secs. 2.1 to 2.3. The basis of the calculation is as follows: if one compares, on one hand, a situation wherein the entire spacecraft is at the same potential as the surface element in question, with, on the other hand, a more realistic situation wherein the rest of the spacecraft is at a less negative potential (Fig. 2.11), then in the latter case, the potential well surrounding the surface element will be steeper and less spatially extended, and the ion collection will in general be

decreased. When $S_i \neq 0$, this argument is subject to qualifications not present in the nondrifting case, for which it is rigorously true in a wide range of conditions (Laframboise and Parker 1973; Laframbose and Godard 1974). In particular, one can envision hypothetical asymmetric sheath potentials which would cause a high-speed-ratio ambient ion distribution to be focused onto the downstream point. We exclude such cases in what follows.

The most extreme example of steepening would be a potential profile which was equal to space potential almost to the spacecraft surface, then fell discontinuously to surface potential. In this limit, the surface element in question would collect just the downstream space-potential current corresponding to the given ion speed ratio. The downstream-point current-density values corresponding to a unipotential sphere at, respectively, the potential of the surface element and space potential may therefore be regarded as upper and lower bounds on the actual current collection at that potential, the upper bound being subject to the above-mentioned qualifications. The resulting values of local floating surface potential may correspondingly be regarded as upper and lower bounds on more realistic values of this quantity. The above-mentioned upper and lower bounds on current correspond, respectively, to the "three-dimensional" and "one-dimensional" velocity-space cutoffs considered in Secs. 2.1 to 2.3 for nondrifting situations.

2.4.2 THEORY OF LOCAL ION COLLECTION ON A UNIPOTENTIAL SPHERE

We assume a collisionless plasma with a drifting Maxwellian ion velocity distribution and negligible magnetic field, containing a fully charge-absorbing, unipotential, spherical electrode. We assume that Debye length $\lambda_D \gg$ electrode radius r_s. In the resulting spherically-symmetric Laplace potential $\phi(r) = \phi_s r_s / r$, the nondimensional ion current density at the electrode surface is (Godard 1975, p. 31)

$$j_i = \int_{\max(0,\chi_s)}^{\infty} \int_0^{\beta-\chi_s} \exp(-\beta - 2S_i\beta^{1/2} \cos\mu \cos\theta - S_i^2)\, I_0(2S_i\beta^{1/2} \sin\mu \sin\theta)\, d\Omega\, d\beta \tag{2.16}$$

where $\chi_s = q\phi_s/kT$, $\beta = E/kT$, $\Omega = L^2/(2mr_s^2 kT)$, $j = J/[N_\infty q(kT/2\pi m)^{\frac{1}{2}}]$, I_o is the modified Bessel function of zero order, N_∞ is number density far from the electrode, μ is angular surface position coordinate measured from the upstream direction, θ is change in direction of the radius vector of a particle as it moves from infinity to radial distance r_s, and θ is related to particle energy E, angular momentum L, charge q and the potential profile $\phi(r)$ by the following expression (Goldstein 1950, Ch. 3):

$$\vartheta = \int_{r_s}^{\infty} L dr/\{r^2[2mE - 2mq\phi(r) - L^2/r^2]\}^{\frac{1}{2}} \tag{2.17}$$

We have computed j_i by integrating Eq. (2.16) numerically. For the given Laplace potential, Eq. (2.17) can be integrated analytically. We obtain:

$$\vartheta = \sin^{-1}[(2\Omega + \chi_s)/(\chi_s^2 + 4\beta\Omega)^{\frac{1}{2}}] - \sin^{-1}[\chi_s/(\chi_s^2 + 4\beta\Omega)^{\frac{1}{2}}] \tag{2.18}$$

For space potential ($\chi_s = 0$), Eq. (2.16) can be integrated analytically. The result is (Tsien 1946)

$$j_i = \pi^{\frac{1}{2}} S_i \cos\mu [1 + \operatorname{erf}(S_i \cos\mu)] + \exp(-S_i^2 \cos^2\mu) \tag{2.19}$$

Figure 2.12 shows results obtained for the ion current density $j_{i\pi}$ at the downstream point $\mu = \pi$, as a function of S_i, with χ_s as a parameter, where $\chi_s = e\phi_s/kT_i \leq 0$ and $e \equiv q_i$. As expected, $j_{i\pi}$ decreases with increasing S_i and increases with increasing $|\chi_s|$. In Figure 2.13 the same results are graphed logarithmically as functions of χ_s. Figure 2.13 shows that these results may be approximated with an error $\lesssim 5\%$ by power-law relations of the form

$$j_{i\pi}(\chi_s) = j_{i\pi}(\chi_s = 0) + A_\pi |\chi_s|^{\alpha_\pi} \; ; \quad \chi_s \leq 0 \tag{2.20}$$

The resulting S_i dependence of the coefficients A_π, α_π and $B_\pi \equiv j_{i\pi}(\chi_s = 0)$ is shown in Fig. 2.14.

2.4.3 RESULTS AND DISCUSSION

Upper and lower bounds on negative downstream-point floating potentials for a shaded, isolated surface element, obtained by numerical solution of the equation $J_i + J_e = 0$, are shown in Fig. 2.15 for various ion-to-electron temperature ratios $\varepsilon = T_i/T_e$. Here we have assumed that ambient electrons are Maxwellian, and that J_i is given alternatively by Eq. (2.20) with Fig. 2.14 and by Eq. (2.19), yielding upper and lower bounds on ion current, corresponding respectively to "three-dimensional" and "one-dimensional" ion velocity-space cutoffs (Sec. 2.4.1). We have also assumed that secondary, backscattered, and photoemitted electron currents are zero. The lower-bound results are subject to the qualifications noted in Sec. 2.4.1. The dashed lines in Fig. 2.15 represent floating potentials for the nondrifting case $S_i = 0$. At ion speed ratios larger than those shown, the situation becomes complicated by electron speed ratio effects, especially at larger values of ε. In Fig. 2.15 we see that at larger values of ε, effects of S_i become important at smaller S_i values.

In Fig. 2.16, upper and lower bounds are shown which are similar to those of Fig. 2.15, except that instead of Maxwellian electron velocity distributions, we have used the "quiet" and "disturbed" electron distributions measured by Shield and Frank (1970) and DeForest and McIlwain (1971) respectively in the Earth's outer magnetosphere, and approximated by Knott (1972), as described in Sec. 2.2; see also Prokopenko and Laframboise (1977) and Laframboise and Prokopenko (1978). The ion temperatures used are 111.6 eV and 2.43 keV, respectively. These values were obtained by integrating the electron velocity distributions to find $N_{e\infty}$, equating $N_{i\infty}$ to the result, then assuming that the ions were Maxwellian and that the ratio of ion to electron random fluxes was 0.025. This procedure differs from that used by Knott (1972) and in Secs. 2.1 - 2.3 in which an ion-to-electron random flux ratio of 0.025 and an ion temperature of 1 keV were assumed simultaneously, thereby violating ambient charge neutrality in general. The corresponding electron mean energies are 270 eV and 8.78 keV. The method used for calculating electron currents is described in Sec. 2.2. We see that S_i effects become important at smaller S_i values in "quiet" magnetospheric conditions. The ratio of ion to electron mean energies implied by the

above data is also larger in "quiet" conditions, corresponding to the dependence of S_i effects on ε noted in Fig. 2.15. The "quiet" and "disturbed" distributions also differ substantially in shape (Knott, 1972, Figs. 1 and 2b). The onset of "significant" drift effects (i.e. floating potential changes $\gtrsim$ 10%) is seen to occur at S_i values as low as 0.1, depending on conditions. It occurs at lower S_i values in the presence of the "quiet" distribution than in any of the other cases shown in Figs. 2.15 and 2.16. In Figs. 2.15 and 2.16, negative floating potentials for $S_i = 1$ are in most cases about twice as large as the corresponding potentials for nondrifting situations.

3. MULTIPLE FLOATING POTENTIALS, AND THE "THRESHOLD MATERIAL TEMPERATURE FOR HIGH-VOLTAGE CHARGING"

3.1 INTRODUCTION

The high-voltage charging behaviour of spacecraft surfaces, especially in outer-magnetospheric plasma conditions, displays a variety of unexpectedly complex features.

These features are most evident in the absence of photoemission (as on shaded surfaces of a spacecraft). One of them is the existence of multiple roots (zeros) in the current-voltage characteristic of various spacecraft materials exposed to certain kinds of ambient plasma environments. An example of such a current-voltage characteristic was shown in Fig. 2.5. In this illustration, only the right- and left-hand roots, which are located at +1.9 and -4100 volts, respectively, represent stable floating surface potentials, because the centre root is an unstable one, in the sense that any voltage excursion from the indicated value (-500V) would result in a net current of a sign which would cause the voltage excursion to increase, ultimately driving the voltage to the right- or left-hand roots.

The possibility of such triple-root situations was first proposed by Whipple (1965, pp. 4-7). Prokopenko and Laframboise (1977, 1980) calculated current-voltage characteristics of various surface materials in outer-magnetospheric (including geostationary-orbit) plasma environments, and found numerous examples in which triple-root characteristics were actually obtained. Sanders and Inouye (1979) did calculations to examine the ranges of conditions in which such characteristics would occur. Besse (1981) examined the mechanisms underlying them. Meyer-Vernet (1982) showed that they may also occur on dust grains in space, such as those in Saturn's rings. It is now generally recognized that in triple-root situations, the stable floating potential near space potential is the result primarily of a current balance between incident "primary" electrons and emitted (secondary and backscattered) electrons,

with incident ion current making only a minor contribution, while at the more negative stable potential, the current balance is primarily between incident ion and electron currents, even though in this case also, both of these may be substantially modified by secondary or backscattered electron emission.

Another feature of the high-voltage charging phenomenon is the occurrence of "sensitivity" effects in the numerical prediction of spacecraft potentials (Stannard et al, 1981), in which relatively small changes in assumed surface properties or ambient conditions can cause large changes in spacecraft floating potentials. Evidently, this phenomenon may frustrate attempts to make reliable predictions of spacecraft charging, and it is important to identify the parameter ranges in which such sensitivity effects occur.

This phenomenon is closely related to that of "threshold" effects, both predicted (Stannard et al, 1981), and observed (Gussenhoven and Mullen, 1982), in which no high-voltage charging occurs over a large range of environmental or surface conditions, but a small further change in conditions then results in a large change in surface potential from a small value (relative to space) to a value typically several kilovolts negative.

Another closely-related effect is that of sudden large changes in surface potential in response to relatively slow temporal changes in ambient plasma conditions (Sec. 2.3). This phenomenon was predicted by Prokopenko and Laframboise (1980) and Besse (1981), and subsequently observed both on the SCATHA (P78-2) satellite and in numerical simulations made using the NASA Charging Analyzer Program (NASCAP) (Schnuelle et al, 1981; Stannard et al, 1982).

In Secs. 3.2 and 3.3, we introduce the concept of threshold temperature as a property of a spacecraft surface material, and we show that all of the above-mentioned phenomena are unified and explained by this concept.

A separate phenomenon, which often controls the differential charging of other surfaces, including sunlit ones, relative to the most highly-charged surface, is the "barrier" effect; we discuss this in Sec. 4.5.

3.2 THE THRESHOLD-TEMPERATURE PROPERTY.

A typical secondary-electron yield curve is shown in Fig. 3.1(a). For most commonly-used spacecraft surface materials, there exists a region of this yield curve in which more than one secondary is produced on average per primary. This generally occurs for incident electron kinetic energies of a few hundred eV. In Fig. 3.1(b), energy-differential incident-electron flux (particle current density) curves are shown for Maxwellian electron velocity distributions at two representative values of electron temperature T_e. In the example shown, the peak of the lower-temperature curve is at almost the same energy as that of the secondary yield curve. In this situation, there is a large production of secondaries; in fact, the total secondary flux, which is given by an integral involving the product of these two functions [Eq. (2.10); Prokopenko and Laframboise (1977, 1980, Eq. 10)], will be <u>greater</u> than the incident primary flux if the maximum secondary yield δ_{max} per primary is greater than about 1.16 (Besse, 1981, Fig. 2). In contrast, the peak of the higher-temperature curve does not coincide closely with that of the secondary-emission curve, and in this case, the total secondary flux will be <u>less</u> than the incident flux. Evidently, a critical value of T_e must exist, intermediate between the two values indicated, at which emitted flux would <u>exactly balance</u> incident flux (H.I. Cohen, 1982, private communication). (There will also be another such critical temperature, below the peak of the secondary yield curve, but this is not of importance here.)

Maxwellian ambient distributions also have the special property that if they are retarded by a repelling (negative) surface potential, the distribution of particle kinetic energies reaching the surface is independent of the value of this potential. Therefore, for a Maxwellian ambient distribution, the preceding conclusions have a special property: they are independent of spacecraft surface potential (Besse, 1981; H.I. Cohen, private communication, 1982) as long as this potential is negative (with respect to space). Therefore, if the ambient electrons were Maxwellian at the critical temperature, incoming and outgoing fluxes would balance each other for all negative values of surface potential, and the surface then "would not know at what potential to float", i.e. the surface potential would become indeterminate, except for the (relatively small) current contribution from ambient ions. It is therefore evident that very large (negative) increases in floating potential will result for very small increases in ambient electron temperature in the neighbourhood of this critical value, which is a property of the surface material only. We will therefore refer to this critical value as the threshold material temperature T* for high-voltage charging. In terms of this property, the first four effects mentioned in Sec. 3.1 can be immediately explained, as follows.

(1) Triple-root current-voltage characteristics have a simple explanation if the ambient electron velocity distribution can be approximated by a double Maxwellian. Such an approximation, although empirically-based, is often a very good one (Garrett and DeForest, 1979 ; Garrett et al, 1981). In this case, a characteristic will have three roots if (i) the temperature T_1 of one Maxwellian is less than T*, but the other one T_2 is greater than T*, and (ii) the total emitted electron (secondary and backscattered) flux at space potential is greater than the incident flux.

This can be proven as follows. The roots of the current-voltage characteristic are given by the zeros of the function:

$$J_{net} = J_i - J_e + J_{sec} + J_{scat} \tag{3.1}$$

where J_i, J_e, J_{sec} and J_{scat} are particle current densities (fluxes) for incident (ambient) ions and electrons, secondary electrons and backscattered electrons, respectively. Except at relatively large negative surface potentials ϕ_s relative to space, J_i is relatively small, and will be neglected in what follows. For $\phi_s < 0$, we then have, in this approximation:

$$J_{net} = (J_{sec,1} + J_{scat,1} - J_{e,1}) \exp(e\phi_s/kT_1)$$

$$+ (J_{sec,2} + J_{scat,2} - J_{e,2}) \exp(e\phi_s/kT_2)$$

$$\equiv J_{net,1} \exp(e\phi_s/kT_1) + J_{net,2} \exp(e\phi_s/kT_2) \tag{3.2}$$

where $e > 0$ is the elementary charge, k is Boltzmann's constant, and all double-subscripted J are space-potential values. A triple-root characteristic must have an unstable root, i.e. a value of ϕ_s such that (a) $J_{net} = 0$, and (b) $dJ_{net}/d\phi_s > 0$. This can happen only for $\phi_s < 0$ because J_{sec} and J_{scat} both decrease rapidly at positive potentials (an exception to this, applicable at very small electron temperatures, has been found by Meyer-Vernet (1982)). Condition (a) requires that $J_{net,1}$ and $J_{net,2}$ have opposite signs, i.e. T^* is between T_1 and T_2. Condition (b) implies that $J_{net} > 0$ for all ϕ_s values between this root and zero; this in turn implies that $J_{net,1} + J_{net,2} > 0$. This completes the proof of (i) and (ii). Even though this proof is only approximate for real velocity distributions, it will still be generally valid in terms of "temperatures" related in the usual way ($\bar{E} = \frac{3}{2} kT$) to the mean energies obtained from a double-Maxwellian fit. Meyer-Vernet (1982) has shown that triple roots can also occur with Maxwellian electrons, when the electron temperature is less than the "temperature" of emission of secondary electrons. The unstable root then o urs at a positive rather than negative value of ϕ_s.

(2) For T(or $2\bar{E}/3k$) close to T^*, most current-voltage characteristics will have both a very small value and small slope over a large range of potentials, and the floating potential(s) will then be subject to large changes when only small changes in conditions occur. This explains "sensitivity" effects.

(3) For T slightly less than T^*, $J_{sec} + J_{scat} > J_e$ if $\phi_p < 0$, and the floating potential will generally be slightly positive. For T slightly greater than T^*, $J_{sec} + J_{scat} < J_e$ and the floating potential will generally be very negative. Clearly T^* is the temperature at which "threshold" effects may be expected.

(4) For a distribution which is (nearly) Maxwellian and has $T \approx T^*$, small (or gradual) changes in ambient conditions can cause the sign of $J_{sec} + J_{scat} - J_e$ to change, resulting in large, sudden changes in floating potential. For a distribution which may not be near-Maxwellian but which leads to a triple-root characteristic, changes in ambient conditions may cause two of the three roots to coalesce and disappear, also producing large, sudden potential changes (Sec. 2.3; Prokopenko and Laframboise, 1980; Besse, 1981; Meyer-Vernet, 1982).

3.3 CALCULATION OF THRESHOLD TEMPERATURES; DISCUSSION

We have calculated threshold-temperature values (Table 2) for a variety of spacecraft surface materials; a listing of our computer program for doing this appears in Appendix B. In order to do this, we have calculate $J_{sec} + J_{scat} - J_e$ as a function of T for Maxwellian incident distributions, and searched numerically for the value T* at which this function changes from positive to negative as T increases.

For ambient electrons normally incident to a surface, we have used the secondary and backscattered flux expressions given by Prokopenko and Laframboise (1977, 1980), together with data given by Dekker (1958), Hachenberg and Brauer (1959, 1962), Gibbons (1966), Willis and Skinner (1973), Katz et al (1977, p. 38), Schnuelle et al (1979), Leung et al (1981), and Krainsky et al (1981).

The increased yields for electrons incident at other angles have an important influence on our results (last three columns of Table 2). For both secondary (Dekker, 1958; Salehi and Flinn, 1981; Krainsky et al, 1981) and backscattered (Darlington and Cosslett, 1972; Krainsky et al, 1981) electrons, it is found experimentally that the dependence of yield (average number of emitted electrons per incident electron) $\delta(E,\theta)$ on angle of incidence θ relative to the surface normal can be usefully approximated by a relation of the form

$$\ln \delta(E,\theta) = \ln \delta(E,0) + \beta(1-\cos\theta). \tag{3.3}$$

For secondary emission, the coefficient β appears to depend primarily on E/E_{max}, where E_{max} is the energy at which δ is largest when $\theta = 0^o$. For backscattered emission, β appears to depend most strongly on the atomic number Z of the surface material, and only weakly on incident energy E. Also, available information on E dependence is fragmentary, and in any case, secondary emission predominates over backscattered emission at smaller values of E. For these reasons, we have ignored the E dependence.

The dependence given by Eq. (3.3) has the special advantage that for any <u>isotropic</u> ambient velocity distribution, the integration over angle in the emitted flux expression can be done analytically. In this case, we have:

$$J_{sec} \text{ or } J_{scat} = \int f\, \delta\, v_n\, d^3\vec{v}$$

$$= \iiint f(E)\delta(E,\theta)\ (v\cos\theta)\ (v^2 \sin\theta\, dv\, d\theta\, d\phi)$$

$$= 2\pi \int_0^\infty f(E)\delta(E,0)v^3 dv \int_0^{\frac{1}{2}\pi} e^{\beta(E)(1-\cos\theta)} \cos\theta \sin\theta\, d\theta \tag{3.4}$$

where (v,θ,ϕ) are spherical coordinates in velocity space with polar axis normal to the surface, $f \equiv d^6N/d^3\vec{r}\ d^3\vec{v}$ is the velocity distribution of ambient electrons, $E = \frac{1}{2}mv^2$, and v_n is the normal component of incident electron velocity.

The integral over θ yields $[\exp(\beta)-\beta-1]/\beta^2$, which has the value $\frac{1}{2}$ when $\beta = 0$. After β is specified as a function of E, multiplication of $\delta(E,0)$ by the factor $\Gamma = (2/\beta^2)[\exp(\beta)-\beta-1]$ then corrects $\delta(E,0)$ to include angle-dependence of the secondary or backscattered yield [see also Whipple, 1981, Eq. (3.11)]. To obtain the resulting dependence of $\delta \equiv \delta_{sec} + \delta_{scat}$ on E, we require that E_{max} and δ_{max} for normal-incidence secondary yield be specified, together with Z. Our complete yield algorithm then is:

$$\xi = \ln(E/E_{max});\quad \eta = 0.2755(\xi - 1.658) - \sqrt{[0.2755(\xi-1.658)]^2+0.0228}\ ;$$

$$\beta_{sec} = e^{\eta};\quad \Gamma_{sec} = (2/\beta_{sec}^{\,2})\left[\exp(\beta_{sec}) - \beta_{sec} - 1\right]; \tag{3.5}$$

$$\beta_{scat} = 7.37Z^{-0.56875};\quad \Gamma_{scat} = (2/\beta_{scat}^{\,2})\left[\exp(\beta_{scat}) - \beta_{scat} - 1\right]$$

where the above expressions for β_{sec} and β_{scat} have been obtained by fits to the data of Dekker (1958), Darlington and Cosslett (1972), Salehi and Flinn (1981), and Krainsky et al (1981). Finally:

$$\delta(E) = \left[7.4\delta_{max} \frac{E}{E_{max}} \exp\left(-2\sqrt{\frac{E}{E_{max}}}\right)\right] \Gamma_{sec} + \left[A-Be^{-CE}\right] \Gamma_{scat} \qquad (3.6)$$

where the factors multiplying Γ_{sec} and Γ_{scat} are, respectively, the secondary-yield curve of Sternglass (1954a), and an empirical factor given by Eq. 2.9 (See also Prokopenko and Laframboise, 1980, Eq. (9)), in which the coefficients A, B, and C are functions of Z obtained from the data of Sternglass (1954b) and Palluel (1947). These coefficients are also displayed in Table 2. $J_{sec} + J_{scat}$ is then given by 2π times the integral over v in (3.4) with f replaced by the Maxwellian distribution corresponding to temperature T, or more generally, by Eqs. (2.10) - (2.14). $J_e = n_\infty(kT/2\pi m_e)^{\frac{1}{2}}$, i.e. the usual random flux for a Maxwellian distribution (where n_∞ is ambient electron density and m_e is electron mass), or more generally, is given by Eqs. (2.1) - (2.3).

The resulting threshold temperatures T* (labeled "TC3") appear in the last column of Table 1. Corresponding values of T* for simplified forms of $\delta(E)$ as indicated (labeled "TC1" and "TC2") appear in the two adjacent columns. It is clear from these results that angle-dependence of δ_{sec} and δ_{scat} has an important effect on values of T*. The values labeled "TC1" can also be inferred from Fig. 2 of Besse (1981).

Clearly, those surface materials having the largest values of T* will be the "most resistant" to high-voltage charging (leaving aside the effects of material conductivity) because the ambient environment will exceed their threshold temperature the least often. From this viewpoint, and using the data of Table 1, MgF_2 is the "most resistant" material, followed by activated beryllium-copper, gold, and NASCAP 'BOOMAT', a spatially-averaged representation of a composite surface consisting of kapton partly covered with platinum strips, used on the SCATHA satellite (Schnuelle et al, 1979). The zero entries for T* in Table 1 are those for which $J_{sec} + J_{scat} < J_e$ at all values of T.

Our discussion so far has been almost completely concerned with "absolute" or "overall" surface charging, and has been based only on calculations of local currents to and from surfaces. Calculations of this type are usually sufficent to determine the floating potential of the most highly (usually negatively) charged portion of a spacecraft surface, which is usually in a shaded or partly-shaded region of the spacecraft. However, the most damaging effects of high-voltage charging are "differential" effects involving large potential differences between adjacent parts of a spacecraft. These effects are frequently dominated by non-local phenomena, an example of which we present in Sec. 4.6.

4. CYLVIA: A TWO-DIMENSIONAL CHARGING SIMULATION FOR CYLINDRICAL SPACECRAFT CROSS-SECTIONS WITH ANGLE-DEPENDENCE.

4.1 INTRODUCTION: THE QUASISTATIC ITERATION METHOD

We have constructed a numerical simulation program called CYLVIA (CYLinder Voltages in Ionosphere and Above) which combines the following features:

a) infinite circular cylindrical geometry with angle-dependence. We have chosen this geometry for the following reasons: A realistic model must be at least two-dimensional because the asymmetry between sunlit and shaded surfaces, or between surfaces with large and small secondary-electron emission currents, is a key feature of the differential charging problem. Cylindrical geometry is the simplest two-dimensional geometry, and it is also a useful approximation to many spacecraft shapes, including the DSCS and SCATHA mainframes, which are finite circular cylinders. Effects of finite cylinder length (three-dimensionality) on sheath potential profiles, which tend to cause a more rapid decrease of potential with radius, can be approximated by using a modified form of Poisson's equation (Sec. 4.3).

b) Quasistatic time-dependent iteration (Laframboise and Prokopenko, 1977). In this procedure, sheath potential changes during particle transit times are ignored. This leads to the following iteration scheme: a distribution of surface potentials is chosen. Poisson's equation is then solved to provide a radius- and angle-dependent static sheath potential [see (c) below]. Particle orbits are then followed numerically in this potential, yielding net surface charging rate as a function of surface position. Using this information, the surface potentials are updated. This process is repeated until a steady-state floating condition is obtained or in order to follow temporal changes in external conditions, such as spacecraft rotation or eclipse passage.

c) use of simplified space-charge density expressi (Sec. 4.2; Laframboise and Prokopenko, 1977), rather than numerical orbit-following,

in solving Poisson's equation for sheath potentials. Approximations of this type are good ones for geostationary-orbit situations, in which the Debye length of the ambient plasma is generally large enough that space-charge effects are almost negligible. They also yield an important saving of computer time.

Features (b) and (c) are now also used in the NASCAP (Katz et al, 1977, 1979) simulation program.

A listing of CYLVIA appears in Appendices C and D. The numerical solution of Poisson's equation in CYLVIA calculations uses the subroutines POIS, POISGN, POINIT, TRID, TRIDP, and NCHECK written by Swarztrauber and Sweet (1975).

4.2 USE OF APPROXIMATE SPACE-CHARGE DENSITY EXPRESSIONS

At geostationary-orbit altitude, the Debye length λ_D for ambient particles is usually $\gtrsim$ 10 m, so for satellites of ordinary size, effects of ambient space charge on sheath potentials will be relatively small. Any reasonably realistic approximation of this space charge can therefore be expected to produce only negligible errors in solving Poisson's equation for sheath potentials. Furthermore, large savings in computer time can be expected to result if one can avoid exact density calculations involving numerical orbit-following. In calculations using CYLVIA, only a relatively small amount of orbit-following is done, in order to calculate surface currents (Sec. 4.4).

A more significant space-charge effect near the spacecraft may be caused by emitted photoelectrons or secondary electrons (Soop, 1972; Schröder, 1973), because of their relatively low velocities compared to ambient values. However, effects of these are likely also to be small enough that any reasonably realistic approximations for their densities will yield good accuracy (Lafon, 1976). Such approximations must ultimately be verified by comparison with exact calculations. It is advantageous if such approximations depend on local potential only (rather than potentials at many locations), together with a relatively small number of other parameters, such as spacecraft potentials and potential barrier heights and locations. Laframboise and Prokopenko (1977) have developed such an approximation. Here we develop three

approximate expressions for ambient space-charge density [Eqs. (4.1)-(4.3)], based on the use of exact density expressions developed for collisionless, Maxwellian particles in the presence of obstacle-free potential wells of arbitrary shape by Laframboise and Parker (1973). The appropriate expression for our purposes is the result given by their Eq. (2) for three-dimensional wells. This is true even for an "infinite", that is, very long cylindrical spacecraft geometry, because of particle entry at the ends of such a geometry. For definiteness, we consider a negative well given by $\phi(x,y,z) \le 0$, with $\phi \to 0$ as $x^2 + y^2 + z^2 \to \infty$, where ϕ is electric potential. If only ambient particles are considered, Poisson's equation is:

$$\nabla^2 \phi = \frac{e}{\varepsilon_o} (N_e - N_i) \tag{4.1}$$

where e is magnitude of unit electron charge, ε_o is permittivity of space, and N_e, N_i are electron and ion number densities, respectively. Since positive ions are the attracted species in this well, we use Eq. (2) of Laframboise and Parker (1973) for ion density, and the usual Boltzmann factor for electron density. If $\lambda_{De} = (\varepsilon_o kT_e/e^2 N_\infty)^{\frac{1}{2}}$, N_∞ is electron or ion density far from the spacecraft, L is a characteristic spacecraft length, $\tilde{\nabla} = L\nabla$, $\chi = e\phi/kT_e < 0$, k is Boltzmann's constant and T is temperature, Eq. (4.1) becomes:

$$\tilde{\nabla}^2 \chi = \left(\frac{L}{\lambda_{De}}\right)^2 \left\{ e^{\chi} - \frac{2}{\sqrt{\pi}} \left[(-\chi\, T_e/T_i)^{1/2} + g\ (-\chi\, T_e/T_i)^{1/2} \right] \right\} \tag{4.2}$$

where $g(s) = \frac{1}{2}\sqrt{\pi}\ \exp(s^2)\mathrm{erfc}(s) = \exp(s^2) \int_s^\infty \exp(-t^2)dt$. An efficient method for calculating erfc(s) has been given by Shepherd and Laframboise (1981).

The important feature of Eq. (4.2) for our purposes is that its right-hand side is a function of χ only. For small χ, Eq. (4.2) reduces to:

$$\tilde{\nabla}^2 \chi = (1 + T_e/T_i)(L/\lambda_{De})^2\, \chi \tag{4.3}$$

where terms of order $\chi^{3/2}$ and higher have been ignored. The linear form of (4.3) permits the use of direct Poisson-solvers for finding χ. Another simplified form can be obtained by rederiving Eq. (4.2) with monoenergetic instead of Maxwellian ions assumed. The appropriate monoenergetic velocity

distribution (Chen, 1965; Laframboise, 1966, p. 14) is:

$$f \equiv \frac{d^3N}{d^3\vec{v}} = \frac{m_i^2 N_\infty}{4\pi} \frac{\delta(E - E_1)}{(2m_i E_1)^{1/2}} \tag{4.4}$$

where $E_1 = 4kT_i/\pi$ and m_i is ion mass; this distribution duplicates the ambient number density and flux values of a Maxwellian at temperature T_i. Rederivation of (2) using this distribution yields the computationally simpler form:

$$\tilde{\nabla}^2 \chi = \left(\frac{L}{\lambda_{De}}\right)^2 \left[e^{\chi} - \left(1 - \frac{\pi}{4}\frac{T_e}{T_i}\chi\right)^{1/2} \right] . \tag{4.5}$$

If any regions exist where $\chi > 0$, the roles of ions and electrons are interchanged, and Eqs. (4.2) - (4.5) must be modified accordingly.

The essential approximation contained in Eqs. (4.2) - (4.5) is the neglect of orbit depletion due to intersection with the spacecraft. The densities of ambient ions and electrons will therefore both be overestimated near the spacecraft in these results. As long as the spacecraft is at least moderately smaller than λ_{De}, the effects of this overestimate will be small. The attracted-species density will be overestimated by the greater amount for reasons involving the curvatures of attracted and repelled particle orbits. The sheath profiles predicted by (4.2) or (4.5) will therefore be steeper than real profiles, if electron emission effects are ignored. Laframboise and Prokopenko (1977, Secs. 2.4.2 and 2.4.3) have also discussed space-charge density approximations based on symmetric potentials and on equivalent potential wells, for ambient and emitted electrons.

4.3 APPROXIMATE INCLUSION OF FINITE-CYLINDER - LENGTH (THREE-DIMENSIONAL) EFFECTS IN POISSON'S EQUATION.

Effects of finite spacecraft length on sheath potential profiles can be included in either of two approximate ways which lead to modifications of the two-dimensional Poisson equation to be solved. We include here a description of these methods. The first method is derived by pretending that the circular inner boundary of the computation grid, which

represents the spacecraft surface, is no longer a cross-section of an infinite cylinder, but rather is a cross-section through the equatorial plane of a prolate spheroid of polar-to-equatorial axis ratio $L \geq 1$. We also assume that the sheath potential is (for some unspecified reason) independent of the latitude coordinate perpendicular to this plane. This leads to a modified Poisson equation of the form

$$\tanh^2 \xi \frac{\partial^2 \chi}{\partial \xi^2} + \tanh \xi \frac{\partial \chi}{\partial \xi} + \frac{\partial^2 \chi}{\partial \theta^2} = \frac{\sinh^2 \xi}{\sinh^2 \xi_s} \left(\frac{R_s}{\lambda_D} \right)^2 (n_e - n_i) \tag{4.6}$$

where $\chi = e\phi/kT_e$, ξ is a radial coordinate in the equatorial plane and is related to nondimensional radius $r = R/R_S$, defined in the same plane, by the relation

$r = (L^2 - 1)^{\frac{1}{2}} \sinh \xi$, $\xi_s = \frac{1}{2} \ln [(L+1)/(L-1)]$, θ is angular coordinate in the same plane, R_S is spacecraft radius, λ_D is Debye length, and n_e and n_i are the nondimensional electron and ion densities N_e/N_{e_∞} and $N_i/N_{i\infty}$, where N_∞ is ambient density of either species. Use of Eq. (4.6) in place of the usual polar-coordinate Poisson equation would result in sheath potential profiles which became increasingly steeper as L decreased, thus allowing for approximate estimates of sheath potentials around finite cylinders. The limiting case $L = 1$ would correspond to an assumed spherical geometry without latitude dependence; the limit $L \to \infty$ leads to recovery of infinite cylindrical geometry.

The transformation $s = \ln \coth \frac{1}{2}\xi$ leads to the alternative form

$$\frac{1}{\cosh^2 \xi} \frac{\partial^2 \chi}{\partial s^2} + \frac{\partial^2 \chi}{\partial \theta^2} = \frac{\sinh^2 \xi}{\sinh^2 \xi_s} \left(\frac{R_s}{\lambda_D} \right)^2 (n_e - n_i) \tag{4.7}$$

which contains no first-order terms. For small ξ, s varies logarithmically with r; for large ξ, s varies as r^{-1}.

The second method is derived by first writing the nondimensional Poisson equation for cylindrical coordinates, which has the form

$$\frac{\partial^2 \chi}{\partial r^2} + \frac{1}{r} \frac{\partial \chi}{\partial r} + \frac{1}{r^2} \frac{\partial^2 \chi}{\partial \theta^2} + \frac{\partial^2 \chi}{\partial z^2} = \left(\frac{R_s}{\lambda_D} \right)^2 (n_e - n_i) \ . \tag{4.8}$$

We then assume that $\chi(r,\theta,z)$ is periodic in z, such that values of χ repeat after nondimensional distance 2ℓ parallel to the z axis. In particular, we assume that $\chi(r,\theta,\ell)$ = some given dependence $\chi_\ell(r,\theta)$,

and that $\chi(r,\theta,0) = \chi_o(r,\theta)$ to be found. We further assume that $\partial\chi/\partial z = 0$ at $z = 0$, $z = \pm\ell$, $z = \pm 2\ell$, etc, and that only the lowest Fourier component of the z dependence of χ is present. Then

$$\chi(r,\theta,z) = \tfrac{1}{2}[\chi_o(r,\theta) + \chi_\ell(r,\theta)] + \tfrac{1}{2}[\chi_o(r,\theta) - \chi_\ell(r,\theta)] \cos\left(\frac{\pi z}{\ell}\right) \quad (4.9)$$

and, at $z = 0$, we have

$$\frac{\partial^2 \chi_o}{\partial z^2} = \tfrac{1}{2}\left(\frac{\pi}{\ell}\right)^2 [\chi_\ell(r,\theta) - \chi_o(r,\theta)] \quad (4.10)$$

The Poisson equation for $\chi_o(r,\theta)$ now becomes

$$\frac{\partial^2 \chi_o}{\partial r^2} + \frac{1}{r}\frac{\partial \chi_o}{\partial r} + \frac{1}{r^2}\frac{\partial^2 \chi_o}{\partial \theta^2} - \tfrac{1}{2}\left(\frac{\pi}{\ell}\right)^2 \chi_o = \left(\frac{R_s}{\lambda_D}\right)^2 (n_e - n_i) - \tfrac{1}{2}\left(\frac{\pi}{\ell}\right)^2 \chi_\ell(r,\theta) \quad (4.11)$$

We see that in this Poisson equation, effects of z-dependence are represented by a homogeneous "Helmholtz" term and a fictitious space charge contribution. The z-dependence incorporated into this equation could represent approximately the effects on sheath potentials of finite spacecraft length and/or features such as conductive circumferential bands. Equations (4.7) and (4.11) are both solvable by standard methods; both are linear. Both contain only two (radius and angle) independent variables.

4.4 ORBIT INTEGRATION AND CURRENT CALCULATION

CYLVIA uses a form of the particle orbit equations in which particle total energy is explicitly conserved. This formulation was adopted because of a difficulty which arose when using more standard methods to integrate photoelectron orbits. Accumulation of numerical errors was occasionally found to change the total energy of an orbit by amounts large compared to the assumed thermal energy of emission (1.5V), especially near points where orbits were "reflected" by a potential barrier; this in turn produced large errors in calculations of photoelectron currents reimpacting spacecraft surfaces.

In order to derive the orbit equations, we consider the motion of a particle in a plane. We let (r,θ) and (v_r, v_θ) represent its position and

velocity components in polar coordinates (Fig. 4.1). We let s represent arc length along its orbit, and $\hat{u}$ and $\hat{n}$ represent unit tangent and unit normal vectors at a point on the orbit, the latter directed toward its local centre of curvature. We let ρ represent its local radius of curvature. We let q,m, and E represent particle charge, mass, and total energy, and $\phi(r,\theta)$ represent electric potential. The equation of motion $m\, d\vec{v}/dt = -q\nabla\phi$ reduces to:

$$v \frac{dv}{ds}\hat{u} + \frac{v^2}{\rho}\hat{n} = \frac{q}{m}\left(-\hat{n}\frac{\partial\phi}{\partial n} - \hat{u}\frac{\partial\phi}{\partial s}\right) \tag{4.12}$$

We equate respective components of Eq. (4.12) and use the relations $ds = \rho\, d(\alpha + \theta)$, $dr = \cos(\alpha)ds$, and $r\, d\theta = \sin(\alpha)ds$. We then obtain the orbit equations in the following form:

$$\begin{aligned}
\frac{d\alpha}{ds} &= -\frac{q}{mv^2}\left(\frac{\cos\alpha}{r}\frac{\partial\phi}{\partial\theta} - \sin\alpha\frac{\partial\phi}{\partial r}\right) - \frac{\sin\alpha}{r} \\
\frac{dr}{ds} &= \cos\alpha \\
\frac{d\theta}{ds} &= \frac{\sin\alpha}{r} \\
v^2 &= \frac{2}{m}\left[E - q\phi(r,\theta)\right]
\end{aligned} \tag{4.13}$$

This system is reduced from fourth to third order because the last equation appears in integrated form. At points where particle reflection from potential barriers produces cusps or near-cusps in an orbit ($d\alpha/ds$ becomes singular or large), a segment of the orbit is replaced by a parabolic arc.

We illustrate the current calculation method used by CYLVIA by first considering photoelectrons which arrive at a point on the surface whose normal makes an angle θ with the sunward direction, each of them having originated at some other surface location θ_o (Fig. 4.2a) and forced to return to the surface by a potential barrier which surrounds the spacecraft. Their current density at the surface location given by θ is:

$$J(\theta) = \int_{v=0}^{v=\infty} \int_{\omega=0}^{\omega=\pi} \hat{f}(v,\omega)(v \sin\omega)(v\, dv\, d\omega) \tag{4.14}$$

where $v = (v_r^2 + v_\theta^2)^{\frac{1}{2}}$ and $\omega = \tan^{-1}(v_r/v_\theta)$ are polar coordinates in incident velocity space at the surface location θ, $\hat{f} \equiv d^2N/dv_r dv_\theta$ is the two-dimensional velocity distribution of photoelectrons, and N is their number density. By Liouville's theorem, $\hat{f}$ is constant along a particle orbit. Assuming that photoelectrons are emitted with a Maxwellian distribution corresponding to a temperature T, their emission flux $J_{ph}(\theta_o)$ is related to $\hat{f}$ as follows:

$$\hat{f} = \frac{1}{\sqrt{2\pi}} J_{ph}(\theta_o) \left(\frac{m}{kT}\right)^{3/2} e^{-mv_o^2/2kT} \tag{4.15}$$

If the sunlit side of the spacecraft has uniform material properties, then

$J_{ph}(\theta_o) = J_{ph}(0) \cos\theta_o$. We introduce dimensionless variables as follows:

$$\chi = q\phi/kT; \qquad u = v(m/2kT)^{\frac{1}{2}} \tag{4.16}$$

Since $\frac{1}{2}mv^2 + q\phi = \frac{1}{2}mv_o^2 + q\phi_o$, (4.15) and (4.14) become:

$$\hat{f} = \frac{1}{\sqrt{2\pi}} J_{ph}(\theta_o)\left(\frac{m}{kT}\right)^{3/2} e^{\chi_o - \chi} e^{-u^2} \tag{4.17}$$

$$J(\theta) = \frac{2}{\sqrt{\pi}} \int_{u=0}^{u=\infty} du\, u^2 e^{-u^2} \int_{\omega=0}^{\omega=\pi} d\omega \sin\omega \left[J_{ph}(\theta_o) e^{\chi_o - \chi}\right] \tag{4.18}$$

The factor in square parentheses in (4.18) is evaluated for each u and ω by integrating the corresponding photoelectron orbit backward to its origin to find θ_o and χ_o. To do the integrations in (4.18) we set up a polar-coordinate grid in velocity space at the surface location θ, as shown in fig. 4.2b, where we have defined $u_n = -u_r$, $u_t = -u_\theta$. We approximate $F(u,\omega) \equiv (2/\sqrt{\pi})\, J_{ph}(\theta_o) \exp(\chi_o - \chi)$ in each cell $u_i \le u \le u_{i+1}$, $\omega_j \le \omega \le \omega_{j+1}$ by $(A + Bu)(C + D\omega)$ where A, ..., D can be determined if the values of F at its four corners are found, again by integrating orbits backward. Equation (4.18) then becomes:

$$J(\theta) = \sum_{i,j} \int_{u_i}^{u_{i+1}} du\, u^2\, e^{-u^2} (A_{ij} + B_{ij}u) \int_{\omega_j}^{\omega_{j+1}} d\omega\, \sin\omega\, (C_{ij} + D_{ij}\omega), \quad (4.19)$$

a form in which all integrals can be evaluated analytically. This method for evaluating $J(\theta)$ is essentially equivalent to the "inside-out" method of Parker and Whipple (1967). The factor $\exp(\chi_o - \chi)$ in $F(u,\omega)$ may vary strongly within individual cells. The potential barrier which surrounds a spacecraft is always of finite height, permitting some photoelectrons to escape and ambient electrons to reach it. This means that the integration in (4.19) must be performed over two regions of velocity space, labelled I and II in Fig. 4.2b, containing photoelectrons (and secondary and backscattered electrons), and ambient electrons, respectively. In general, $\hat{f}$ will contain a discontinuity at the boundary between I and II (Whipple, 1976) which can produce large errors in the evaluation of $J(\theta)$. The integration method used in CYLVIA treats these discontinuities explicity, using bisection searches to find points such as those circled in Fig. 4.2b. If the ambient electron velocity distribution is isotropic, then F in region II will be independent of ω.

4.5 RESULTS AND DISCUSSION

Figure 4.3 shows a CYLVIA calculation of equipotential contours surrounding a cylindrical spacecraft cross-section whose surface consists of two independently floating conductive sectors, the smaller of which is shaded and subtends an angle of 90°. In this calculation the ambient ion and electron velocity distributions are double Maxwellians with the following properties:

$N_{i1} = 1\ cm^{-3}$	$N_{e1} = 1\ cm^{-3}$
$T_{i1} = 20\ eV$	$T_{e1} = 500\ eV$
$N_{i2} = 1\ cm^{-3}$	$N_{e2} = 1\ cm^{-3}$
$T_{i2} = 10^4\ eV$	$T_{e2} = 5000\ eV$

The photoelectron charge flux eJ_{ph} is $45 \times 10^{-6} A/m^2$ at normal sunlight incidence. $T_{ph} = 1.5$ eV. Secondary and backscattered electron fluxes are assumed zero. Ambient ion and electron and photoelectron currents are calculated using numerical orbit-following as described in Sec. 4.4. The computation grid in (r,θ) contains 65×48 intervals. In this and subsequent calculations, the computation grid in (υ, ω) contains 8×16 intervals for each Maxwellian component of each particle species, apart from bisection searches (Sec. 4.4) which give finer resolution. Linear space charge is assumed [Sec. 4.2; Laframboise and Prokopenko, 1977, Eq. (3)]. The above-mentioned plasma parameters imply an ambient Debye length of 32.5 meters; spacecraft radius r_s is 1 meter. The outer boundary of the computation grid is at $e^5 r_s \approx 148 r_s$. The most noteworthy feature of Fig. 4.3 is a negative saddle-point potential barrier which surrounds the larger sector, and whose height varies from about 2 volts at the sunward point to several hundred volts near the edges of this sector. The potential of the 270° sector (-2.265 kV) is controlled by blocking of electron escape caused by this barrier; this mechanism is discussed in more detail in Sec. 4.6. In Sec. 6.5, surface photocurrents obtained from this calculation are compared with values given by an analytic approximation. In this and subsequent calculations, calculated surface potentials are all within 10 volts of the values at which the residual currents change sign.

Figure 4.4 shows a CYLVIA calculation of equipotential contours around a spacecraft cross-section divided into four 90° sectors, two covered with aluminum and two with quartz. The aluminum sectors are electrically connected to each other. In this calculation the ambient velocity distributions are single Maxwellians with $N_i = N_e = 3cm^{-3}$ and $T_i = T_e = 1$ keV. Photoelectron fluxes at normal sunlight incidence are 42 and 32 $\mu A/m^2$ for aluminum and quartz, respectively. $T_{ph} = 1.5$ eV for both. Secondary and backscattered fluxes are assumed zero. The computation grid in (r,θ) contains 65×16 intervals. The outer boundary of this grid is at $e^7 \approx 1097$ spacecraft radii. The above-mentioned plasma parameters imply an ambient Debye length of 96 meters; spacecraft radius r_s is 1 meter. As in Fig. 4.3, a saddle-point barrier is present on the sunward side of the spacecraft (see Sec. 4.6), and controls the potential of the sunlit quartz sector and the two aluminum sectors.

In a separate calculation, which is not shown, we included secondary and backscattered electron emission; the calculated surface potentials then were all between 0 and +10V. This result can be readily interpreted in terms of the threshold material temperatures for high-voltage charging defined in Sec. 3 and listed in Table 2. Although the value of T_e for this case is greater than the threshold temperature of aluminum (Table 2), the aluminum sectors are partly sunlit, and photoemit enough to balance their ambient electron collection current. The threshold temperature of quartz is greater than T_e, so secondary and backscattered electron emission from these sectors is therefore enough to balance their ambient electron collection current. Thus in this case, no negative charging occurs on any sector. If T_e were made substantially larger than the threshold temperature of quartz (Table 2), then high-voltage negative charging of the shaded quartz sector would again occur.

In Fig. 4.5, the same spacecraft cross-section as in Fig. 4.4 has been rotated by 90° relative to the sunward direction. In this case the most negative potentials are those on the shaded regions of the quartz sectors, which float separately from the sunlit regions because the quartz is nonconductive. In this situation it is possible that

breakdown of orbit-limitation of the collection of ambient ions (Sec. 2.1) could occur, especially near the edges of these shaded regions, and this would restrict ion collection and cause their floating potentials to become even more negative than we have calculated, but the rather coarse θ-interval of our computational grid has prevented us from resolving this question in this calculation.

Also in Fig. 4.5, we have this time a saddle-point barrier on the shaded side of the spacecraft; if the calculation had included secondary and backscattered electrons, this barrier would have controlled their escape. In this case a negative current flows from the shaded to the sunlit aluminum sector, so the sunlit sector floats at a much more negative potential than it would otherwise.

Figure 4.6 shows the distribution of surface potentials corresponding to the calculations of Figs. 4.4 and 4.5, together with the calculated surface potential distribution of a completely nonconductive ("quartz" but with secondary and backscattered electron emission not included) cylinder which is placed in the same environment.

Figure 4.7 shows again the same surface potential distribution for a nonconductive cylinder as in Fig. 4.6, together with another distribution for conditions which are unchanged except that the outer boundary r_B of the calculation is moved inward to 12 spacecraft radii r_S. A third calculation is shown, which has been done not using CYLVIA but using a program called TWOD, which combines the physical assumptions of the NASCAP program with circular cylindrical geometry (M. Mandell, Systems, Science and Software Inc., private communication). Except as noted, the physical situations treated in the CYLVIA and TWOD calculations are the same; in both cases $r_B = 12\ r_S$, zero secondary and backscattered emission is assumed, and a 16-point angular discretization is used. In the TWOD calculation, zero space charge is assumed, but since $\lambda_D = 96\ r_S$ in the CYLVIA calculation, the comparison shown is probably unaffected by this

difference. In the TWOD calculation, photoelectron temperature and normal-incidence flux are 2 eV and 20 $\mu A/m^2$, whereas for CYLVIA, the corresponding parameters are 1.5 eV and 45 $\mu A/m^2$.

We see that rough agreement exists between the CYLVIA and the TWOD calculations. Either our higher photoemission flux, or the rather coarse angular discretization used in both calculations, may account for the differences; we have not yet investigated this question with further calculations. Even though the barrier which surrounds the sunlit side of the cylinder prevents almost all photoelectron escape in the (two) regions $60^o \lesssim \theta < 90^o$ where the largest disagreement occurs, our larger assumed flux permits greater surface migration of photoelectrons (Sec. 6), and the resulting surface current will drain excess negative charge from these regions (to an extent which we have so far not determined). The large difference between the two CYLVIA calculations for $r_B = 12r_S$ and $r_B = e^7\ r_S$ is symptomatic of the great sensitivity which two-dimensional Laplace-potential (or nearly Laplace-potential, as in our case) calculations have to outer-boundary position generally. On the other hand, tests with CYLVIA have indicated that $r_B \gtrsim e^5\ r_S$ is sufficient to overcome this sensitivity, especially when a small amount of linear (or other) space charge is included, as we have done.

4.6 THE BARRIER EFFECT

As we have seen (Secs. 2 and 3), charging calculations based on local-current-balance considerations are usually sufficient to determine the floating potential of the most highly (usually negatively) charged portion of a spacecraft surface, which is usually in a shaded or partly-shaded region of the spacecraft. However, the most damaging effects of high-voltage charging are "differential" effects involving large potential differences between adjacent parts of a spacecraft. These effects are frequently dominated by non-local phenomena, several examples of which occurred in Sec. 4.5, and which we now examine in more detail.

The most important among these phenomena is the "barrier" effect, which often controls the differential charging of other surfaces, including sunlit ones, relative to the most highly-charged surface. In the "barrier" effect, a strong dipole or higher moment of the spacecraft's potential distribution produces "saddle-point" barriers over less-highly-charged portions of the spacecraft, limiting photoemitted or other electron escape from these portions and causing them to charge more negatively than otherwise. Space-charge effects are normally not important in the formation of such barriers. Their existence, and the resulting implications for differential charging, were first predicted by Fahleson (1973). Whipple (1976) presented evidence for the existence of such a barrier on the ATS-6 satellite. Other properties and consequences of these barriers were discussed by Prokopenko and Laframboise (1977,1980) and Laframboise and Prokopenko (1977). Katz et al (1979) performed a numerical simulation which showed the formation of such a barrier near a polyhedral "quasispherical" model satellite, together with the resulting effects for differential charging. Besse and Rubin (1980) developed an analytical treatment of the barrier effect for a spherical satellite. Purvis (1982) presented a variety of NASCAP numerical simulations to illustrate the prevalence of the barrier effect in high-voltage charging situations. Katz and Mandell (1982) examined mechanisms underlying the barrier effect, called by them the "field-reversal" effect, and in one particular application, the "snapover" effect. In this Section, we present results of a numerical simulation for circular cylindrical geometry, which emphasize the importance of the barrier effect for differential charging.

Using CYLVIA (Secs. 4.1 - 4.5), we have calculated the equilibrium charging state of a cylindrical spacecraft shape with a nonconductive surface in a model geostationary-orbit plasma with sunlight incident normally to the cylinder axis on one side. Figure 4.8 shows the results of a CYLVIA calculation in which we have deliberately made an important oversimplification: we have calculated the surface potential distribution on the basis of <u>local current balance only</u>. Because sunlight

strikes only one side of the spacecraft, photoelectrons are emitted only from this side. In this example, the escaping photoelectron flux from most of the sunlit side is sufficient to balance the ambient electron flux, so the sunlit side of the spacecraft floats at a potential of 5.1V, while the shaded side floats at -2.96 kV. As a result, the spacecraft potential has a strong dipole moment, resulting in the existence of a potential barrier which surrounds its sunlit side. The height of this barrier is -0.61 kV, while the assumed photoelectron emission temperature is only 1.5 eV, so almost all photoelectrons reflect from the barrier and return to the spacecraft, an effect which we have not taken into account in this calculation.

In Fig. 4.9 we have included this effect, and have allowed CYLVIA to converge to the resulting more-realistic steady state. As a result of photoelectron reflection from the barrier, the sunlit side of the spacecraft has charged to a more negative potential. As a result, the barrier height in the sunward direction has been reduced to -0.2V. A substantial fraction of the photo-electrons now escape over the barrier, permitting the local currents to come into balance. A further indication of this is that the saddle point has moved inward from 5.9 spacecraft radii almost to the spacecraft surface: such an inward movement was first predicted by Fahleson (1973), and is evident also in the results of Katz et al (1979, Figs. 17-21) and Besse and Rubin (1980).

The resulting changes in the potential profile along the spacecraft-sun line are shown in Fig. 4.10. The inward motion of the saddle point is again evident. Most importantly, the amount of differential charging between the sunlit and shaded sides has been reduced dramatically, from 2.96 kV to 1.19 kV. Potential-barrier formation evidently exerts a controlling effect on the differential charging. This is increasingly recognized as being the normal condition in differential-charging situations (Purvis, 1982; Katz and Mandell, 1982). Because the capacitances between various parts of a spacecraft are generally much larger than the free-space capacitance of the entire spacecraft, the

charging times for differential charging are usually much larger than those for absolute charging (Purvis, 1982).

It is clear from our example and also the examples presented in Sec. 4.5 (and also another example to be presented in Sec. 5.2), that the formation of potential barriers is very geometry-dependent. This is the case because in geostationary-orbit conditions, space-charge shielding around spacecraft of "ordinary" size (up to a few tens of meters) is a small effect (since the ambient Debye length is usually at least this large), and electric fields produced by charging on one part of a spacecraft surface readily surround other parts of it , even if these are on the other side of the spacecraft. Detailed simulations which include realistic representation of spacecraft geometry therefore appear likely to remain important for studies of spacecraft charging whenever the possibility of barrier effects exists.

In Fig. 4.11 we show another CYLVIA calculation which is discussed in detail in Sec. 5.

5. XYCIC: A SIMULATION FOR GENERAL PLANAR-SYMMETRIC (TWO-DIMENSIONAL) SPACECRAFT GEOMETRIES

5.1 INTRODUCTION

We have developed and partly tested a program called XYCIC [(X,Y) Charging Investigation Code], which is designed to permit simulation of a wider class of two-dimensional geometries than does CYLVIA. XYCIC is designed to treat any spacecraft cross-section which takes the form of one or more polygons, each of which joins a set of unit lattice points in a Cartesian plane with lines having slopes of 0, $\pm\frac{1}{2}$, ± 1, ± 2, or ∞. Thus a circular cylinder can be approximated by either an octagon or a hexadecagon (16-sided polygon). At present, the geometric features, or "object-generation" portion of XYCIC, and its Poisson-solver, have been completed and tested, and program segments incorporating the same plasma simulation features as in CYLVIA (Secs. 4.1 - 4.4) have been written and partly tested. A listing of the present version of XYCIC appears in Appendix E.

As noted in connection with Fig. 4.7, plasma simulations in two dimensions generally involve strong sensitivity to boundary effects, unless space-charge shielding intervenes, an effect which frequently does not occur at the large Debye lengths typical of geostationary-orbit conditions. In such cases, as we have seen, it may become necessary to place the outer boundaries of computational domains very far from a (simulated) spacecraft. In order to accomplish this in XYCIC without the penalty of excessive numbers of grid points, we have constructed its computational domain as a set of nested square grids, each of which is centered in the next larger one, and such that in crossing the boundary from each to the next larger one, the grid interval is doubled. A similar succession of nested sub-domains is used in three dimensions in NASCAP (Katz et al, 1977). On the boundary of the outermost domain, the potential is assumed to be zero.

5.2 PRELIMINARY RESULTS

Figure 5.1 shows a XYCIC calculation of equipotential contours around an octagonal approximation to the circular cylindrical geometry of Fig. 4.11. As in the case of Fig. 4.11, the surface potentials in Fig. 5.1 have the imposed values shown, rather than self-consistent values. A total of 7 nested grids have been used in this calculation, with the outermost grid boundary located at 27.4 half-widths of the simulated object.

This calculation, and the CYLVIA calculation shown in Fig. 4.11, have been done for comparison with a NASCAP calculation (Olsen, 1980, p. 190; Olsen and Whipple, 1980, Fig. 16) of potentials around an octagonal cylinder "model object" which approximates the ATS-5 satellite. The feature of greatest interest in both sets of calculations is the potentials of the four saddle points which occur outside the 50 V sections of the spacecraft surface, which represent conductive areas from which electron emission occurred on the real spacecraft. Comparison of our results, Figs. 4.11 and 5.1, with each other indicates a satisfactory level of agreement between them with regard to the potentials and locations of the saddle points. This is the case even though the CYLVIA calculation contains a small amount of linear space charge, while the XYCIC calculation contains none.

Comparison of either result with that of Olsen and Whipple indicates that our saddle points are located about twice as far from the spacecraft surface as theirs, and have larger negative potentials (-56 to -59 V) than theirs (-53 V). These differences undoubtedly result from the fact that our simulation is two-dimensional and theirs is three-dimensional, even though our geometry in Fig. 5.1 is identical with that of their cross-section. Incorporation of three-dimensionality into our calculation, using the approximate method described in Sec. 4.3, would probably bring our results into much closer agreement with theirs, but we have not yet done this. In modifying our calculation in this fashion, it would probably be advantageous to perform our calculation as the superposition

of <u>two</u> modified calculations of the type described by Eqs. (4.8) to (4.11). One of these would have surface potentials given by the (uniform) <u>average</u> (taken over surface position) of those in our Figures, and the characteristic length ℓ used in it would be that of the ATS-5 model object. The other would have surface potentials given by the <u>departures</u> from this average, and its (much smaller) value of ℓ would be that of the conductive patches on the ATS-5 object. The disturbance potential of the patches would then decrease more quickly with radius, in better agreement with the Olsen and Whipple calculation.

Figure 5.2 shows a XYCIC prediction of equipotentials around a composite "object" which represents a cross-section through a hypothetical spacecraft-body-and-antenna combination. Again the surface potential values are given ones; in this case they are hypothetical. Other data pertinent to this calculation are given in the figure caption. If self-consistent calculations are made in the future with a geometry similar to the one shown, the "cutout" in the spacecraft body, and the region between the spacecraft body and the antenna, would both become examples of the "shaded cavities" discussed in connection with Fig. 2.8. It would then be possible to verify the prediction, made in Sec. 2.3, that electrically-isolated surfaces inside such cavities may charge to larger negative potentials than those elsewhere on the spacecraft. It would also be possible to investigate the relative importance of the two mechanisms, discussed in Sec. 2.3, which may produce such charging.

6. AN ANALYTIC CALCULATION OF SURFACE PHOTOCURRENTS

6.1 INTRODUCTION

Photoelectron migration can be an important cause of surface currents on spacecraft in charging situations. Numerical methods of calculating photoelectron migration involve following a large number of electron orbits which will generally be short (from origin to impact point) and have large curvature. Such a procedure can be a major source of expense in operating a simulation program. A good analytic approximation for surface photocurrent can therefore be of great value. If photoelectron migration takes place over a curved surface, the sunlight incidence angle will vary over this surface and therefore so will the photoemission flux. On the other hand, if the normal component of electric force on the electrons is attractive toward the surface and is large enough, the total distance of their travel along the surface will be short enough that effects of surface curvature on their orbits can be neglected, so a model situation involving a planar surface with a photoemission gradient along it becomes appropriate. In Section 6.2, we perform an analytic calculation of surface photocurrent for such a situation. In Section 6.3, we do a partial numerical verification of our result by comparing it with a numerical result obtained using CYLVIA.

6.2 THEORY

In this Section, we derive an analytic expression for the surface current density of photoelectron migration along a plane surface $y = 0$, in the presence of: (a) a uniform normal electric field $E_y > 0$, which causes photoelectrons emitted from the surface to reimpact it (Fig. 6.1) (b) a uniform tangential electric field E_x (c) a uniform photoemission current density gradient $J'_{ph} = dJ_{ph}/dx$, so that the photoemission current per unit surface area is $J_{ph}(x) = J_{ph,o} + J'_{ph}\, x$. This photoemission gradient, or "production gradient", would ordinarily be caused by a spatial variation in the illumination of the surface. We also assume

that photoelectrons are emitted with a Maxwellian velocity distribution corresponding to a temperature $T = T_{ph}$. In the presence of (a) and (b), all photoelectron orbits are parabolas whose axes are parallel to the resultant electric field vector (Fig. 6.1). The impact location x for a photoelectron which originates at x_o with emission velocity components v_{xo} and v_{yo} is:

$$x = x_o + \frac{2m}{eE_y}\left(v_{xo}v_{yo} - \frac{E_x}{E_y}v_{yo}^2\right) \tag{6.1}$$

where e is the magnitude of unit electronic charge. The surface current Γ in the x direction, per unit distance z perpendicular to the (x,y) plane, can now be found by integrating over position and velocity of emission to find the number of photoelectrons per unit z and unit time which cross the plane x = 0 in the direction of increasing x, then subtracting the corresponding result for decreasing x. We note that $v_{yo}\,\hat{f}(x_o, v_{xo}, v_{yo})$ is the number of photoelectrons produced per unit surface area per unit v_{xo} and v_{yo}, where $\hat{f} \equiv d^2N/dv_x dv_y$ is the two-dimensional velocity distribution of photoelectrons and N is their number density. We obtain:

$$\Gamma = \int_{-\infty}^{0} dx_o \int_{-\infty}^{\infty} dv_{xo} \int_{0}^{\infty} dv_{yo} v_{yo}\,\hat{f}(x_o, v_{xo}, v_{yo}) H_+(x_o, v_{xo}, v_{yo})$$
$$- \int_{0}^{\infty} dx_o \int_{-\infty}^{\infty} dv_{xo} \int_{0}^{\infty} dv_{yo}\, v_{yo}\,\hat{f}\,(x_o, v_{xo}, v_{yo})\, H_-\,(x_o, v_{xo}, v_{yo}) \tag{6.2}$$

where $\hat{f} = (1/2\pi) J_{ph}(x)\,(m/kT)^{3/2} \exp(-mv_o^2/2kT)$, and H_+ and H_- are equal to 1 if the impact location x given by (6.1) is positive or negative, respectively, and equal to 0 otherwise. We define:

$$v_o^2 = v_{xo}^2 + v_{yo}^2\ ;\ \psi = \tan^{-1}(v_{yo}/v_{xo})\ ;\ \xi = \tan^{-1}(E_y/E_x)\,. \tag{6.3}$$

Since $v_{yo} > 0$, (6.1) implies that $x \gtrless x_o$ if $v_{xo} \gtrless (E_x/E_y)v_{yo}$, or $\psi \gtrless \xi$.

Equation (6.2) now becomes:

$$\Gamma = \frac{1}{\sqrt{2\pi}}\left(\frac{m}{kT}\right)^{\frac{3}{2}} \int_0^{\infty} dv_o v_o^2 \exp\left(-\frac{mv_o^2}{2kT}\right) \int_0^{\xi} d\psi\, \sin\psi \int_{-\frac{2mv_o^2}{eE_y}(\cos\psi\sin\psi - \cot\xi\,\sin^2\psi)}^{0} dx_o\,(J_{ph,o} + J'_{ph}\,x_o)$$
$$- \frac{1}{\sqrt{2\pi}}\left(\frac{m}{kT}\right)^{\frac{3}{2}} \int_0^{\infty} dv_o\, v_o^2\, \exp\left(-\frac{mv_o^2}{2kT}\right) \int_{\xi}^{\pi} d\psi\, \sin\psi \int_0^{-\frac{2mv_o^2}{eE_y}(\cos\psi\sin\psi - \cot\xi\,\sin^2\psi)} dx_o\,(J_{ph,o} + J'_{ph}\,x_o) \tag{6.4}$$

We define:

$$\Gamma_o = J_{ph,o}\, kT/eE_y; \quad \gamma = \Gamma/\Gamma_o;$$
$$\tilde{x} = (eE_y/kT)x; \quad j' = J'_{ph}\,\Gamma_o/J_{ph,o}^{\;2}; \tag{6.5}$$
$$u = v_o\sqrt{m/2kT}\ .$$

Then (6.4) becomes:

$$\gamma = \frac{2}{\sqrt{\pi}}\int_o^\infty du\ u^2\ e^{-u^2}\int_o^\pi d\psi\ \Big[4u^2(\cos\psi\sin^2\psi - \cot\xi\sin^3\psi)$$
$$- 8u^4 j'(\cos^2\psi\sin^3\psi - 2\cos\psi\sin^4\psi\cot\xi + \cot^2\xi\sin^5\psi)\Big]$$
$$= -4\cot\xi - j'(4+16\cot^2\xi)\,. \tag{6.6}$$

Using (6.3) and (6.5), we finally obtain:

$$\Gamma = -\frac{4J_{ph,o}kT_{ph}E_x}{eE_y^{\;2}} - J'_{ph}\left(\frac{kT_{ph}}{eE_y}\right)^2\left[4 + 16\left(\frac{E_x}{E_y}\right)^2\right]\ . \tag{6.7}$$

This result contains, respectively, a potential-gradient term, a production-gradient term, and a cross-term. The effect of the cross-term can be substantial: we see that it enhances the production-gradient term five-fold if $(E_x/E_y)^2 = 1$, in comparison with its value when $E_x = 0$. This is true regardless of the sign of E_x; in other words, surprisingly, an "opposed" electric field causes the same enhancement as an "aligned" one.

The potential-gradient term is twice that given by Eq. (14) of Pelizzari and Criswell (1978); this can be seen as follows. If one assumes that photoelectrons are emitted isotropically, corresponding to a value of zero for the parameter b in their Eqs. (13) and (14), then in our notation, their Eq. (14) is:

$$-e\Gamma = \frac{4J_{ph,o}(\frac{3}{2}kT_{ph})\ E_x}{3E_y^{\;2}} \tag{6.8}$$

where our quantities $-e\Gamma$, $J_{ph,o}$, and $\frac{3}{2}kT_{ph}$ are the same as their J_a, F, and $\langle E \rangle$, respectively. It is readily seen that (6.7) with $J'_{ph} = 0$ gives a result twice as large as (6.8), as just mentioned. To see why this difference occurs, we rederive Pelizzari and Criswell's result more rigorously, as follows. From our Eq. (6.1), the contribution of E_x to particle displacement in the x direction is

$$\frac{-2mE_x\, v_{yo}^2}{eE_y^2} \tag{6.9}$$

which is the same as their Eqs. (9) and (11) combined. The contribution due to v_{xo} in (6.1) averages to zero. Now the surface current density Γ per unit z equals the integral over v_{xo} and v_{yo} of: photoelectron production rate per unit v_{xo} and v_{yo} per unit surface area in the (x,z) plane, times distance travelled by particles of a given v_{yo} before impact [given by (6.9)]. This production rate in turn equals number per unit volume per unit v_{xo} and v_{yo}, times the value of v_{yo}. Therefore:

$$\begin{aligned}\Gamma &= \int_0^\infty dv_{yo} \int_{-\infty}^\infty dv_{xo}\, \hat{f}\, v_{yo} \left(- \frac{2mE_x\, v_{yo}^2}{eE_y^2}\right) \\ &= \frac{J_{ph,o}}{\sqrt{2\pi}} \left(\frac{m}{kT}\right)^{\frac{3}{2}} \left(- \frac{2mE_x}{eE_y^2}\right) \int_{-\infty}^\infty dv_{xo} \exp\left(-\frac{mv_{xo}^2}{2kT}\right) \int_0^\infty dv_{yo}\, v_{yo}^3 \exp\left(- \frac{mv_{yo}^2}{2kT}\right) \\ &= - \frac{4J_{ph,o}\, kT_{ph}}{eE_y^2}\end{aligned} \tag{6.10}$$

in agreement with <u>our</u> result [Eq. (6.7)] rather than that of Pelizzari and Criswell. The reason for the difference evidently involves the fact that we have integrated Eq. (6.9) over v_{yo} but they did not. Thus we perform an integral containing v_{yo}^3 [in the first line of (6.10)], but their procedure involves essentially an integral containing v_{yo}^2 (to evaluate their quantity $\langle E \rangle$), times a separate integral containing v_{yo} (to obtain $J_{ph,o}$ from $\hat{f}$). Their procedure therefore involves the use of incorrect moments of the velocity distribution function of photoemitted electrons.

6.3 COMPARISON WITH A NUMERICAL RESULT FROM CYLVIA

Figure 4.3 shows a CYLVIA calculation of equipotential contours surrounding a cylindrical spacecraft cross-section whose surface consists of two independently floating conductive sectors, the smaller of which is shaded and subtends an angle of 90^{o}. Data pertinent to this calculation are given in Sec. 4.5. The most noteworthy feature of Fig. 4.3 is a negative saddle-point potential barrier which surrounds the larger sector, and whose height varies from about 2 volts at the sunward point to several hundred volts near the edges of this sector.

The resulting normalized current densities j_i, j_e, and j_{ph} of ambient ions, ambient electrons, and photoelectrons are shown as functions of surface position in Fig. 6.2. We have made a separate calculation of j_{ph} using Eq. (6.7) with the tangential electric field E_x set equal to zero since the spacecraft surfaces are conductive. To use (6.7), we note that the net photoelectron flux out of the surface is equal to the divergence of Γ with respect to surface coordinates. In our geometry, this means that

$$J_{ph,net\ in} \equiv J_{ph,in} - J_{ph,out} = \frac{-1}{r_s}\ \frac{d\Gamma}{d\theta} = \frac{4}{r_s^2}\frac{d}{d\theta}\left[\left(\frac{kT_{ph}}{eE_r}\right)^2 \frac{dJ_{ph,out}}{d\theta}\right] \quad (6.11)$$

where

$$e\,dJ_{ph,out}/d\theta = -45 \times 10^{-6}\ \sin\theta\ \ A/m^2 \left(-\tfrac{1}{2}\pi < \theta < \tfrac{1}{2}\pi\right),$$

and the radial electric field E_r is obtained from the numerical solution for $\phi(r,\theta)$ used to construct Fig. 4.3. Net photoelectron currents obtained in this way are shown as dashed curves in Fig. 6.2. We see that near $\theta = 0^{o}$, the net outward photocurrent is badly underestimated by Eq. (6.11) since the potential barrier for electrons at this location is not much higher than the photoelectron mean thermal energy, so a substantial fraction of photoelectrons escape, and this is not allowed for in Eqs. (6.7) and (6.11). However, in the interval $30^{o} \lesssim \theta \leq 90^{o}$, where photoelectron escape is negligible, agreement between Eq. (6.11) and the numerical result is much better. The numerical result is about 10% to 20% above that given by (6.11); the most important reason for this

difference is probably the fact that the tangential electric field, although *zero* at the spacecraft surface, is *nonzero* outside it, and the form of the cross-term in (6.7) indicates that the production-gradient current [which is the one calculated in Eq. (6.11)] is strongly sensitive to such fields. We have shown the photoelectron current as decreasing to zero almost discontinuously beyond $\theta = 90^{\circ}$, because the average angular distance of photoelectron migration in the electric fields at this point ($E_{\theta} = 0$, $E_r = 1824$ V/m) is about 0.1°.

Another noteworthy feature of Fig. 6.2 is the decrease in the flux of ambient electrons at larger θ, caused by the increasing height of the potential barrier as one moves away from the sunward point $\theta = 0^{\circ}$.

7. FLUX AND DENSITY CALCULATION FOR COLLISIONLESS PARTICLE ORBITS

Calculation of current density deposited on a surface by particle orbits neighbouring a given orbit is of importance in several contexts related to spacecraft charging. Examples of these are: calculation of current density deposited on one part of a spacecraft by photoelectrons, secondary electrons, or beams of charged particles emitted from another point on it, or calculation of ion current density deposited on spacecraft surfaces in the presence of either a (model) infinite or a large but finite ion speed ratio. In this Section, a simple, general procedure is described for obtaining such information essentially as a byproduct of a numerical orbit calculation. The procedure is based on a perturbation of the orbit equations, and involves finding the evolution along an orbit of the axes of a differential tube of neighbouring orbits. The positions of these axes are given in terms of a set of integrals, contributions to which are collected as the orbit is followed numerically, and whose integrands involve the space derivatives $\partial F_i/\partial x_j$ of the force components of points along it. At a surface impingement point, current density is then obtained by projecting the cross-section of this tube onto the surface tangent plane. The same formulation can also be used to obtain information about particle number density along an orbit.

In order to appreciate the usefulness of such a procedure it is instructive to compare it with the "standard" procedure which one would normally follow in calculating values of current density deposited on surfaces, and space charge density, of collisionless ions with negligible thermal motion flowing past a collecting object. The "standard" procedure is to numerically follow sets of neighbouring, initially-parallel particle orbits inward from an assumed unperturbed region far from the object, and to calculate flux and density everywhere between any two orbits by finding out how far apart they have become at their impingement points or elsewhere. This method has inherent difficulties: if the orbits chosen are too far apart initally, an orbit may eventually go off in a very different direction than the orbit next to it, making calculations of flux or density between them impossible or unrealistic. On the other hand, if they are initially too close together, inaccuracies in calculating either orbit may obscure the small difference which one is trying to calculate.

Our calculation proceeds as follows. The equations of particle motion are:

$$\frac{d^2\vec{x}}{dt^2} = \frac{F(\vec{x},t)}{m} \tag{7.1}$$

which is equivalent to:

$$\frac{d^2 x_i}{dt^2} = \frac{F_i(x_1,x_2,x_3,t)}{m} \tag{7.2}$$

Integration of this yields:

$$x_i(t) = x_i(0) + v_i(0)t + \int_o^t dt' \int_o^{t'} dt'' \, \frac{F_i[x_k(t''), t'']}{m} \tag{7.3}$$

For perturbed orbits, (7.2) is replaced by:

$$\frac{d^2}{dt^2}(x_i + \delta x_i) = \frac{F_i + \delta F_i}{m} \tag{7.4}$$

Subtraction of (7.2) from (7.4) yields:

$$\frac{d^2}{dt^2}\delta x_i = \frac{\delta F_i}{m} \tag{7.5}$$

We now assume that perturbations at any instant are small. Then:

$$\delta F_i = \sum_j \frac{\partial F_i}{\partial x_j}\delta x_j \tag{7.6}$$

i.e. we ignore $(\delta x_j)^2$, $(\delta x_j)^3$, etc.

Equations (7.5) and (7.6) now imply, after integrating twice:

$$\delta x_i(t) = \frac{1}{m}\int_o^t dt' \int_o^{t'} dt'' \sum_j \frac{\partial F_i}{\partial x_j}[x_k(t''),t'']\delta x_j(t'') + \delta x_i(0) + t\,\delta v_i(0) \tag{7.7}$$

where the nine quantities $\partial F_i/\partial x_j$ must be provided along the unperturbed orbit given by $x_k(t'')$ and t''.

By noting that the region of integration in (7.7) is a triangle in (t',t") coordinates, and then interchanging the order of integration, we can change (7.7) into the equivalent form (A.D. Stauffer, private communication):

$$\delta x_i(t) = \frac{1}{m}\int_0^t dt''(t-t'') \sum_j \frac{\partial F_i}{\partial x_j}[x_k(t''),t'']\delta x_j(t'') + \delta x_i(0) + t\,\delta v_i(0) \tag{7.8}$$

which contains only a single integration.

We now consider a <u>monokinetic</u> particle beam having initial velocity $(v_o,0,0)$. This implies a zero initial perturbation velocity: $\delta\vec{v}(0) = (0,0,0)$. We choose two mutually orthogonal initial perturbations in position:

$$\delta_2\vec{x}(0) = (0,1,0) \tag{7.9}$$

$$\delta_3\vec{x}(0) = (0,0,1)$$

Since (7.8) is both linear and homogeneous in $\delta\vec{x}$, final values of $\delta_2\vec{x}$ and $\delta_3\vec{x}$ are proportional to initial values, as one expects for perturbation quantities.

Furthermore, if $\delta\vec{x}(0) = a\,\delta_2\vec{x}(0) + b\,\delta_3\vec{x}(0)$, then $\delta\vec{x}(t) = a\,\delta_2\vec{x}(t) + b\,\delta_3\vec{x}(t)$, where a and b are <u>unchanged</u>.

We now consider an initially circular differential tube of orbits defined by (7.9). As the unperturbed orbit is followed numerically, we simultaneously calculate $\delta_2\vec{x}$ and $\delta_3\vec{x}$ using (7.8). At the impingement point of the orbit, we project these onto the surface tangent plane (Fig. 7.1). The area of the ellipse thus generated is inversely proportional to the current density deposited on the surface.

For a monokinetic beam, this gives current density <u>with no approximations</u>.

The linearity of the perturbation equation (7.8) now implies that all orbits passing through A_o (Fig. 7.1) also pass through A. Therefore the total current carried by the tube is a constant, equal to $A_o j_o = \frac{\pi}{4} j_o$, where j_o is the current density through A_o.

The area A is given by:

$$A = \frac{\pi}{4} \left| \delta_2 \vec{x} \times \delta_3 \vec{x} \right| = \frac{\pi}{4} \left\| \begin{matrix} \hat{i} & \hat{j} & \hat{k} \\ \delta_2 x_1 & \delta_2 x_2 & \delta_2 x_3 \\ \delta_3 x_1 & \delta_3 x_2 & \delta_3 x_3 \end{matrix} \right\| \tag{7.10}$$

where $\hat{i}$, $\hat{j}$, and $\hat{k}$ are unit vectors along the coordinate axes.

The current density at the surface therefore varies inversely as the projected tube area given by:

$$\frac{\pi}{4} \hat{n} \cdot (\delta_2 \vec{x} \times \delta_3 \vec{x}) = \frac{\pi}{4} \left\| \begin{matrix} n_1 & n_2 & n_3 \\ \delta_2 x_1 & \delta_2 x_2 & \delta_2 x_3 \\ \delta_3 x_1 & \delta_3 x_2 & \delta_3 x_3 \end{matrix} \right\| \tag{7.11}$$

evaluated at the orbit impingement point on the surface, where $\hat{n}$ is a unit vector perpendicular to the surface tangent plane (Fig. 7.1).

The current density at the surface therefore is:

$$j_{surface} = j_o \Bigg/ \left\| \begin{matrix} n_1 & n_2 & n_3 \\ \delta_2 x_1 & \delta_2 x_2 & \delta_2 x_3 \\ \delta_3 x_1 & \delta_3 x_2 & \delta_3 x_3 \end{matrix} \right\|_{surface} \tag{7.12}$$

It is possible for the plane of the ellipse to be perpendicular to the surface tangent plane. This occurs when the perturbed orbits happen to cross each other at the surface. The current density at the surface will then be infinite. This can occur only for a monokinetic distribution of initial velocities, which is only an approximation to real distributions.

We can also obtain an expression for number density n along the orbit, since $n\vec{v}$ times the projection of A on a plane perpendicular to $\vec{v}$ remains constant along the orbit.

i.e. the product $\frac{\pi}{4} n \left| \vec{v} \cdot (\delta_2\vec{x} \times \delta_3\vec{x}) \right|$ is constant.

We therefore obtain:

$$n = n_o v_o \Bigg/ \left\| \begin{matrix} v_1 & v_2 & v_3 \\ \delta_2 x_1 & \delta_2 x_2 & \delta_2 x_3 \\ \delta_3 x_1 & \delta_3 x_2 & \delta_3 x_3 \end{matrix} \right\| \tag{7.13}$$

For a monokinetic distribution, n can also become infinite if the perturbed orbits cross each other. For the same distribution, n(and $j_{surface}$ if a surface is present) can also become infinite if one of the vectors $\delta_2\vec{x}$ or $\delta_3\vec{x}$, or some linear combination of them, becomes zero. An example of this occurs on the (wake) axis of symmetry behind a sphere in a flowing collisionless plasma containing infinite-speed-ratio ions.

In two dimensions the perturbation produces a differential strip rather than tube. The corresponding results are:

$$j_{surface} = j_o \Bigg/ \left\| \begin{matrix} n_1 & n_2 \\ \delta x_1 & \delta x_2 \end{matrix} \right\|_{surface} \tag{7.14}$$

$$n = n_o v_o \Bigg/ \left\| \begin{matrix} v_1 & v_2 \\ \delta x_1 & \delta x_2 \end{matrix} \right\| \tag{7.15}$$

We can also consider velocity-space rather than position-space initial perturbations. We assume that:

$$\delta_2\vec{x} = \delta_3\vec{x} = 0, \quad \text{but} \quad \delta_2\vec{v} = (0,1,0), \text{ and } \delta_3\vec{v} = (0,0,1). \tag{7.16}$$

This permits us to treat situations where the ion speed ratio is large but not infinite. We can use (7.16) to do a calculation of ion defocusing on the wake axis behind a sphere, valid to order S_i^{-1}, where S_i is the ion speed ratio. We consider the situation shown in Fig. 7.2, in which an unperturbed orbit, initially parallel to the ion drift direction,has already been computed, in the presence of some known or given electric potential distribution e.g. a Coulomb potential. This orbit crosses the z axis at a downstream point z_o. Because

all orbits having the same impact parameter r cross at z_o, the ion density at this point is "infinite" unless the ions also have some random motion (S_i is finite). In reality the resulting space charge along this axis will influence ion orbits near it, but we ignore this effect here. We assume that values of $\partial F_r/\partial r$, $\partial F_r/\partial z$, $\partial F_z/\partial r$, and $\partial F_z/\partial z$ are known along the unperturbed orbit. Our procedure is as follows. We choose two initial velocity perturbation vectors $\delta_a\vec{v}(0)$ and $\delta_b\vec{v}(0)$ in the plane of the unpertubed orbit (Fig. 7.2). We also choose $\delta_2\vec{x}(0) = \delta_3\vec{x}(0) = 0$ at z_o. We now do our perturbation calculations <u>backward</u>, or "inside-out" along the ion orbit, even though we initially calculated the orbit forward, in the ion direction of motion. A <u>single</u> integration of (7.5) with (7.6) yields:

$$\delta v_i(t) = \frac{1}{m}\int_o^t dt' \sum_j \frac{\partial F_i}{\partial x_j}[x_k(t'),t']\delta x_j(t') + \delta v_i(0) \; . \tag{7.17}$$

For each of our two initial velocity perturbation vectors, we first integrate (7.8) from z_o to an upstream point at which the orbit is essentially no longer affected by the electric field of the sphere. We then use the resulting values of $\delta x_j(t')$ [in this case $\delta r(t')$ and $\delta z(t')$] in (7.17) to calculate $\delta v_i(t)$ [in this case $\delta v_r(t)$ and $\delta v_z(t)$] at the same upstream point. The linearity of the perturbation calculation
already discussed, implies now that multiplying $\delta_a\vec{v}(0)$ and $\delta_b\vec{v}(0)$ by any constants a and b means that $\delta_a\vec{v}(t)$ and $\delta_b\vec{v}(t)$ are also multiplied by a and b. Suppose now that the velocity distribution at the upstream point is a drifting Maxwellian. The linearity just mentioned now implies that velocity components in the (r, z) plane at the upstream point map linearly into those on the axis at z_o, so an integration over velocity space to find the density at z_o is easy to perform. This now will be found to yield a <u>finite</u> rather than infinite result, essentially because most ions will now have a small amount of angular momentum about the axis of symmetry and will no longer be found at z_o, and our perturbation calculation implicitly takes this into account.

As a final example, we consider the case of a nearly-monokinetic beam (the usual case for ion or electron guns on a spacecraft). We first need to calculate the <u>five</u> perturbation quantities $\delta_2\vec{x},\ldots,\delta_6\vec{x}$ resulting from choosing initial perturbations given by (7.9) and by $\delta_1\vec{v}(0) = (1,0,0)$, $\delta_2\vec{v}(0) = (0,1,0)$, $\delta_3\vec{v}(0) = (0,0,1)$. The values of $\delta_2\vec{x}$ and $\delta_3\vec{x}$ then give, as before, current density or space-change density at any point of interest, but these are now <u>differential</u>

quantities with respect to the velocity distribution $f \equiv d^3N/dv_1dv_2dv_3$ of the beam at its emission point; i.e. the values of these quantities found in this way are proportional to $f(v_1,v_2,v_3)dv_1dv_2dv_3$. The remaining perturbation quantities $\delta_4\vec{x}$, $\delta_5\vec{x}$, and $\delta_6\vec{x}$ then give the required information about the spread of the beam in position space. This may itself be due or partly due to beam space-charge; if this is the case, then the force derivatives $\partial F_i/\partial x_j$ must themselves be found self-consistently, complicating the problem; in this case, calculations of the kind presented here would presumably become part of an iterative scheme.

If the beam is not nearly monokinetic (as will often be the case if it has not been deliberately accelerated) or it creates its own potential barrier around the spacecraft, so that part of it escapes and part does not, then evidently differential perturbations in velocity do not provide a realistic description of the result, and one would then need to replace calculations of $\delta_4\vec{x}$, $\delta_5\vec{x}$, and $\delta_6\vec{x}$ by separate, non-perturbation, orbit calculations for a representative discrete sample of the particle velocities most strongly represented in the beam.

REFERENCES

Allen, J.S., 1939, The emission of secondary electrons from metals bombarded with protons, Phys. Rev. 55, 336.

Axford, W.I., 1968, Observations of the interplanetary plasma, Space Sci. Rev. 8, 331-365.

Baragiola, R.A., Alonso,E.V., Florio, A.O.,1979, Electron emission from clean metal surfaces induced by low-energy light ions, Phys. Rev. B19, 121.

Bernstein, I.B. and Rabinowitz,I.N., 1959, Theory of electrostatic probes in a low-density plasma, Phys. Fluids 2, 112.

Besse, A.L. 1981, Unstable Potential of Geosynchronous Spacecraft, J. Geophys. Res. 86(A4), 2443-2446.

Besse, A.L. and Rubin, A.G.,1980, A Simple Analysis of Spacecraft Charging Involving Blocked Photoelectron Currents, J. Geophys. Res. 85 (A5), 2324-2328.

Chen, F.F., 1965, Numerical computations for ion probe characteristics in a collisionless plasma, Plasma Phys. (J. Nucle. Ener. Part C) 7,47-68.

Chung, M.S., and Everhart, T.E.,1974, Simple calculation of energy distribution of low-energy secondary electrons emitted from metals under electron bombardment, J. Appl. Phys. 45, 707.

Cousinie, P., Colombié, N.,Fert,C.and Simon, R., 1959, Variation du coefficient d'émission électronique secondaire de quelques métaux avec l'énergie des ions incidents, Compt. Rend. 249, 387.

Darlington, E.H. and Cosslett, V.E. 1972, Backscattering of 0.5 - 10keV electrons from solid targets, J. Phys. D: Appl. Phys. 5, 1969- 1981.

DeForest, S.E., 1972, Spacecraft charging at synchronous orbit, J. Geophys. Res. 77, 651.

DeForest, S.E., 1977, The plasma environment at geosynchronous altitude, in: Proceedings of the Spacecraft Charging Technology Conference, edited by C.P. Pike and R.R. Lovell, pp. 37-52, Air Force Surveys in Geophysics, No. 364, Hanscom Air Force Base, Mass., Rep. AFGL-TR-77-0051/NASA TMX-73537.

DeForest, S.E., and McIlwain, C.E., 1971, Plasma clouds in the magnetosphere, J. Geophys. Res. 76, 3587-3611.

Dekker, A.J. 1958, Secondary Electron Emission, in: Solid State Physics, edited by F. Seitz and D. Turnbull, vol. 6, Academic, New York, p.251.

Dessler, A.J., 1967, Solar wind and interplanetary magnetic field, Rev. Geophys. 5, 1-4`.

Fahleson, U. 1973, Plasma-Vehicle Interactions in Space - Some Aspects on Present Knowledge and Future Development, in: Photon and Particle Interaction with Surfaces in Space, edited by R.J.L. Grard, D. Reidel, Dordrecht, Holland, pp. 563-569.

Garrett, H.B. and DeForest, S., 1979, An Analytical Simulation of the Geosynchronous Plasma Environment, Planet.Space Sci. 27, 1101-1109.

Garrett, H.B., Schwank, D.C. and DeForest, S.E., 1981, A Statistical Analysis of the Low-Energy Geosynchronous Plasma Environment - I. Electrons, Planet.Space Sci. 29, (10), 1021-1044.

Gibbons, D.J., 1966, Secondary Electron Emission, in: Handbook of Vacuum Physics, edited by A.H. Beck, Pergamon, Oxford, p. 301.

Godard, R., 1975, A symmetrical model for cylindrical and spherical collectors in a flowing collisionless plasma, Ph.D. Thesis, York University, Toronto.

Goldstein, H., 1950, Classical Mechanics, Addison-Wesley Pub. Co., Reading, Massachusetts.

Goldstein, R. and Divine, N., 1977, Plasma distribution and spacecraft charging modeling near Jupiter. In: Proc. USAF-NASA Spacecraft Charging Technology Conference, C.P. Pike and R.R. Lovell, Editors, Report No. AFGL-TR-77-0051, Air Force Geophysics Laboratory, Hanscom AFB, Massachusetts/NASA TMX-73537, Lewis Research Center, Cleveland Ohio, pp. 131-141.

Gussenhoven, M.S. and Mullen, E.G., A "Worst Case" Spacecraft Charging Environment as Observed by SCATHA on 24 April 1979, paper No. 82-0271, Amer. Inst. Aeron. Astron. 20th Aerospace Sciences Mtg., Jan. 82 , Orlando, Florida.

Hachenberg, O. and Brauer, W., 1959, Secondary Electron Emission from Solids, Advan. Electron.Electron Phys. 11, 413.

Hachenberg, O. and Brauer, W., 1962, Advan. Electron.Electron Phys., 16, 145.

Katz, I., Parks, D.E., Mandell, M.J., Harvey, J.M., Brownell, D.H. Jr., Wang, S.S. and Rotenberg, M., 1977, A Three Dimensional Dynamic Study of Electrostatic Charging in Materials, NASA Contractor Report No. CR-135256, Lewis Research Center, Cleveland, Ohio.

Katz, I., Parks, D.E., Wang, S., and Wilson, A., 1977, Dynamic modeling of spacecraft in a collisionless plasma. In: Proc. USAF-NASA Spacecraft Charging Technology Conference, C.P. Pike and R.R. Lovell, Editors, Report No. AFGL-TR-77-0051, Air Force Geophysics Laboratory, Hanscom AFB, Massachusetts/NASA TMX-73537, Lewis Research Center, Cleveland, Ohio, pp. 319-330.

Katz, I., Cassidy, J.J., Mandell, M.J., Schnuelle, G.W., Steen, P.G., and Roche, J.C., 1979, The Capabilities of the NASA Charging Analyzer Program, in: Spacecraft Charging Technology - 1978, NASA Conference Publication 2071/Report No. AFGL-TR-79-0082, Air Force Geophysics Laboratory, Massachusetts, pp. 101-122.

Katz, I., and Mandell, M.J., 1982, Differential Charging of High-Voltage Spacecraft: The Equilibrium Potential of Insulated Surfaces, J. Geophys. Res. 87, 4533-4541.

Kazan, B., and Knoll,M., 1968, Electronic Image Storage, Academic Press, New York.

Knott, K., 1972, The equilibrium potential of a magnetospheric satellite in an eclipse situation, Planet. Space Sci. 20, 1137.

Krainsky, I., Lundin, W., Gordon, W.L. and Hoffman, R.W., 1981, Secondary Electron Emission Yields, in: Spacecraft Charging Technology 1980, NASA Conference Publication 2182/Report No. AFGL-TR-81-0270, Air Force Geophysics Laboratory, Massachusetts, pp. 179-197.

Lafon, J.-P.J., 1976, On the sheath surrounding a conductor emitting photoelectrons in an isotropic collisionless plasma, Radio Science 11 ,483-493.

Laframboise, J.G., 1966, Theory of spherical and cylindrical Langmuir probes in a collisionless, Maxwellian plasma at rest, Univ. of Toronto, Institute for Aerospace Studies, UTIAS Rep. 100.

Laframboise, J.G., and Parker, L.W., 1973, Probe design for orbit-limited current collection, Phys. Fluids 16, 629-636.

Laframboise, J.G., and Godard, R., 1974, Perturbation of an electrostatic probe by a spacecraft at small speed ratios, Planet. Space Sci. 22, 1145-1155.

Laframboise, J.G. and Prokopenko, S.M.L., 1977, Numerical Simulation of Spacecraft Charging Phenomena, in: Proc. Spacecraft Charging Technology Conference, C.P. Pike and R.R. Lovell, Eds., Report No. AFGL-TR-77-0051, Air Force Geophysics Laboratory, Massachusetts/Report No. NASA TMX-73537, Lewis Research Center, Cleveland, Ohio, pp. 309-318.

Laframboise, J.G., and Prokopenko, S.M.L., 1978, Predictions of high-voltage differential charging on geostationary spacecraft, in: Proc. 1978 Symposium on the Effect of the Ionosphere on Space and Terrestrial Systems, paper 4.4, Naval Research Lab./Office of Naval Research, Washington.

Leung, M.S., Tueling, M.B. and Schnauss, E.R., 1981, Effects of Secondary Electron Emission on Charging, in: Spacecraft Charging Technology 1980, NASA Conference Publication 2182/Report No. AFGL-TR-81-0270, Air Force Geophysics Laboratory, Massachusetts, pp. 163-178.

Manka, R.H., 1973, Plasma and potential at the lunar surface. In: Photon and Particle Interactions with Surfaces in Space, R.J.L. Grard, Editor, D. Reidel Pub. Co., Dordrecht, Holland, pp. 347-361.

Mauk, B., 1975, Magnetospheric substorm pitch angle distribution, EOS 56, 423.

McPherson, D.A., and Schober, W.R., 1976, Spacecraft charging at high altitudes: the Scatha satellite program, in: Spacecraft Charging by Magnetospheric Plasmas, edited by A. Rosen, American Institute of Aeronautics and Astronautics, New York, and MIT Press, Cambridge, Massachusetts.

Meyer-Vernet, N., 1982, "Flip-Flop" of Electric Potential of Dust Grains in Space, Astron. Astrophys. 105, 98-106.

Mott-Smith, H., Jr., and Langmuir, I., 1926, The theory of collectors in gaseous discharges, Phys. Rev. 28, 727.

Olsen, R.C., 1980, Differential and Active Charging Results from the ATS Spacecraft. Univ. of California, San Diego, Ph.D. Thesis.

Olsen, R.C. and Whipple, E.C., 1980, Analysis of Differential and Active Charging Phenomena on ATS-5 and ATS-6. University of California, San Diego, Center for Astrophysics and Space Sciences, Rep. CASS-80-1.

Palluel, P., 1947, Composante rediffusée du rayonnement électronique secondaire des métaux, C.R. Acad. Sci. 224, 1492.

Parker, L.W., 1978, Potential barriers and asymmetric sheaths due to differential charging of nonconducting spacecraft. Report No. AFGL-TR-78-0045, Air Force Geophysics Laboratory, Hanscom AFB, Massachusetts.

Parker, L.W., and Whipple, E.C, Jr., 1967, Theory of a Satellite Electrostatic Probe. Ann. Phys. 44, 126-161.

Pelizzari, M.A., and Criswell, D.R., 1978, Differential Photoelectric Charging of Nonconducting Surfaces in Space. J. Geophys. Res. 83, 5233-5244.

Polychronopulos, B., 1973, Effects of non-Maxwellian electron energy distributions on the orbital limited current-voltage characteristics of cylindrical and spherical Langmuir probes under collisionless conditions, Plasma Phys. 15, 37.

Prokopenko, S.M.L. and Laframboise, J.G., 1977, Prediction of Large Negative Shaded-Side Spacecraft Potentials, in: Proc. Spacecraft Charging Technology Conference, C.P. Pike and R.R. Lovell, Eds., Report No. AFGL-TR-77-0051, Air Force Geophysics Laboratory, Massachusetts/Report No. NASA TMX-73537, Lewis Research Center, Cleveland, Ohio, pp. 369-387.

Prokopenko, S.M.L. and Laframboise, J.G., 1980, High-Voltage Differential Charging of Geostationary Spacecraft, J. Geophys. Res. 85, (A8), 4125-4131.

Purvis, C., 1982, Evolution of Spacecraft Charging Technology, Paper No. 82-0273, Amer. Inst. Aeron. Astron. 20th Aerospace Sciences Mtg., Jan. 82, Orlando, Florida.

Ray, J.A., and Barnett,C.F., 1971, Secondary electron emission of metals bombarded with 120-eV to 5-keV protons, J. Appl. Phys. 42, 3260.

Rosen, A., 1975, Large Arcs and Discharges on Spacecraft, Astronaut. Aeronaut., 13, 36.

Salehi, M. and Flinn, E.A., 1981, Dependence of Secondary-Electron Emission from Amorphous Materials on Primary Angle of Incidence, J. Appl. Phys. 52 (2), 994-996.

Sanders, N.L. and Inouye, G.T., 1979, Secondary Emission Effects on Spacecraft Charging: Energy Distribution Considerations, in: Spacecraft Charging Technology - 1978. NASA Conference Publication 2071/ Report No. AFGL-TR-79-0082, Air Force Geophysics Laboratory, Massachusetts, pp. 747-755.

Schnuelle, G.W., Parks, D.E., Katz, I., Mandell, M.J., Steen, P.G., Cassidy, J.J. and Rubin, A., 1979, Charging Analysis of the SCATHA satellite, in: Spacecraft Charging Technology - 1978, NASA Conference Publication 2071/Report No. AFGL-TR-79-0082, Air Force Geophysics Laboratory, Massachusetts, pp. 123-143.

Schnuelle, G.W., Stannard, P.R., Katz, I. and Mandell, M.J., 1981, Simulation of Charging Response of SCATHA (P78-2) Satellite, in: Spacecraft Charging Technology 1980, NASA Conference Publication 2182/Report No. AFGL-TR-81-0270, Air Force Geophysics Laboratory, Massachusetts, pp. 580-591.

Schröder, H., 1973, Spherically symmetric model of the photoelectron sheath for moderately large plasma Debye lengths. In: Photon and Particle Interactions with Surfaces in Space, R.J.L. Grard, Editor, D. Reidel Pub. Co., Dordrecht, Holland, pp. 51-58.

Shepherd, M.M., and Laframboise, J.G., 1981, Chebyshev approximation of $(1+2x)\exp(x^2)\mathrm{erfc}(x)$ in $0 \leq x < \infty$. Math. of Computation 36, 249.

Shield, M.A., and Frank, L.A., 1970, Electron observations between the inner edge of the plasma sheet and the magnetosphere, J. Geophys. Res. 75, 5401-5414.

Soop, M., 1972, Report on photo-sheath calculations for the satellite GEOS, Planet. Space Sci. 20, 859-870.

Stannard, P.R., Schnuelle, G.W., Katz, I., and Mandell, M.J., 1981, Representation and Material Charging Response of GEO Plasma Environments, in: Spacecraft Charging Technology 1980, NASA Conference Publication 2182/Report No. AFGL-TR-81-0270, Air Force Geophysics Laboratory, Massachusetts , pp. 560-579.

Stannard, P.R., Katz, I., Gedeon, L., Roche, J.C., Rubin, A.G. and Tautz, M.F., 1982, Validation of the NASCAP Model Using Spaceflight Data, Paper No. 82-0269, Amer. Inst. Aeron. Astron. 20th Aerospace Sciences Mtg., Jan. 82, Orlando, Florida.

Sternglass, E.J., 1954, Theory of Secondary Electron Emission, Sci. Pap. 1772, Westinghouse Res. Lab., Pittsburgh, Pa.

Sternglass, E.J., 1954, Backscattering of Kilovolt Electrons from Solids, Phys. Rev. 95, 345-358.

Swarztrauber, P., and Sweet, R., 1975, Efficient FORTRAN Subprograms for the Solution of Elliptic Partial Differential Equations, Tech. Note NCAR-TN/IA-109, National Center for Atmospheric Research, Boulder, Colorado.

Thomas, S., and Pattinson, E.B., 1970, Range of electrons and contribution of back-scattered electrons in secondary production in aluminum, J. Phys. D: Appl. Phys. 3, 349.

Tsien, H.S., 1946, Superaerodynamics, mechanics of rarefied gases, J. Aero. Sci. 13, 653-664.

Whipple, E.C., Jr., 1965, The Equilibrium Electric Potential of a Body in the Upper Atmosphere and in Interplanetary Space, Ph.D. Thesis, The George Washington Univ., Washington, D.C./Report No. X-615-65-296, NASA Goddard Space Flight Center, Greenbelt, Maryland.

Whipple, E.C. Jr., 1976, Observation of Photoelectrons and Secondary Electrons Reflected from a Potential Barrier in the Vicinity of ATS 6, J. Geophys. Res. 81 (4), 715-719.

Whipple, E.C., 1981, Potentials of Surfaces in Space, Rep. Prog. Phys. 44, 1197-1250.

Willis, R.F., and Skinner, D.K., 1973, Secondary electron emission yield behaviour of polymers, Solid State Comm. 13, 685.

Table I

Floating potentials of shaded surfaces of geostationary-altitude spacecraft, using the incident velocity spectra assumed by Knott [1972], with backscattered and secondary electron emission due to electron impacts included, with three-, two-, and one-dimensional velocity-space cutoffs corresponding to orbit-limited ion collection in spherical, infinite cylindrical and planar symmetries, respectively.

Material	Secondary Electron Emission Data		Backscatter Data	Spectrum 1 "Quiet"			Spectrum 2b "Disturbed"		
	δ_{max}	E_{max} (eV)	η(5000eV)	Floating Potential (volts) 3-dimen.	2-dimen.	1-dimen.	Floating Potential (volts) 3-dimen.	2-dimen.	1-dimen.
Gold	1.45	800	.42	-39.9	-40.0	-40.8	-3470	-6430	-15,450
Aluminum	.97	300	.16	-1410	-2140	-5390	-6770	-11,500	-21,770
Aluminum with Oxide Coating	2.60	300	.12	+4.6 -630* -750	+3.0 -490* -1560	+1.9 -420* -4900	-6610	-11,360	-21,610
Quartz	2.50	420	.12	+3.9	+2.5	+1.5 -640* -4120	-6310	-10,960	-21,130
Aquadag	.75	350	.08	-1560	-2380	-5890	-7090	-12,010	-22,350
Beryllium Copper	2.20	300	.31	+4.7	+2.9	+1.9 -560* -3430	-5740	-9920	-19,890
Beryllium Copper Activated	5.00	400	.31	+8.0	+5.8	+4.2	+4.3 -900* -3950	+2.9 -830* -7580	+1.6 -770* -17,670
Teflon	3.00	300	.10	+5.2	+3.6 -660* -1370	+2.3 -490* -4870	-6640	-11,430	-21,700
Kapton	2.10	150	.07	+3.9 -170* -1580	+2.5 -170* -2440	+1.4 -170* -6040	-7180	-12,130	-22,520
No secondary or backscattered electrons	—	—	—	-1860	-2830	-6680	-7550	-12,690	-23,130

* Unstable

TABLE 2

THRESHOLD TEMPERATURES T* FOR SPACECRAFT SURFACE MATERIALS
TC1:INCLUDING SECONDARY ELECTRONS.
TC2:INCLUDING SECONDARY AND BACKSCATTERED ELECTRONS.
TC3:SAME AS TC2 EXCEPT THAT ANGULAR DEPENDENCE OF YIELD IS INCLUDED.

	SECONDARY		BACKSCATTERING				T*(keV)		
MATERIAL	E_{max}(keV)	δ_{max}	Z	A	B	C	TC1	TC2	TC3
GOLD	.80	1.45	79.0	.4802	.3566	.6103	1.278	2.931	4.969
ALUMINUM	.30	.97	13.0	.1568	.0303	.3431	0.000	0.000	.601
ALUMINUM OXIDE	.30	2.60	10.0	.1238	.0172	.3435	1.116	1.777	1.931
SIO2(QUARTZ)	.42	2.50	10.0	.1238	.0172	.3435	1.496	1.713	2.621
FUSED SILICA	.33	3.46	10.0	.1238	.0172	.3435	1.645	1.847	2.707
AQUADAG(COLLOIDAL GRAPHITE)	.35	.75	6.0	.0800	0.0000	0.0000	0.000	0.000	0.000
BERYLLIUM-COPPER	.30	2.20	29.0	.3136	.0692	.6207	.918	1.349	2.157
BERYLLIUM-COPPER(ACTIVATED)	.40	5.00	29.0	.3136	.0692	.6207	2.764	3.692	5.423
TEFLON	.30	3.00	8.0	.0900	0.0000	0.0000	1.299	1.477	2.102
KAPTON (Willis and Skinner, 1973)	.15	2.10	5.3	.0700	0.0000	0.0000	.433	.474	.769
KAPTON (Leung et al, 1981)	.25	1.80	5.3	.0700	0.0000	0.0000	.583	.647	1.096
INDIUM OXIDE	.80	1.40	24.4	.2750	.0600	.5400	1.184	2.010	3.596
MAGNESIUM	.25	.92	12.0	.1460	.0250	.3440	0.000	0.000	.441
MAGNESIUM OXIDE	.40	4.00	10.0	.1238	.0172	.3435	2.280	2.548	3.686
SILVER	.80	1.00	47.0	.3900	.2890	.6320	0.000	1.233	2.754
ITO ON KAPTON	.35	2.55	15.3	.1830	.0370	.3820	1.275	1.550	2.370
IO ON 'FEP' TEFLON	.36	2.39	16.2	.1920	.0400	.3990	1.217	1.507	2.329
ITO ON BOROSILICATE GLASS	.35	2.35	16.9	.2000	.0420	.4100	1.160	1.455	2.255
ION-SPUTTERED ITO ON KAPTON	.39	1.52	15.3	.1830	.0370	.3820	.684	.921	1.587
MGF2	.85	6.38	10.0	.1238	.0172	.3435	7.141	7.889	11.054
NASCAP 'NPAI'	.15	2.10	5.0	.0600	0.0000	0.0000	.433	.468	.747
NASCAP 'SOLA'	.41	2.05	10.0	.1238	.0172	.3435	1.147	1.338	2.123
NASCAP 'CPAI'	.15	2.10	5.0	.0600	0.0000	0.0000	.433	.468	.747
NASCAP 'WHITEN'	.15	2.10	5.0	.0600	0.0000	0.0000	.433	.468	.747
NASCAP 'BLACKC YELOWC'	.15	2.10	5.0	.0600	0.0000	0.0000	.433	.468	.747
NASCAP 'GOLDPD'	.72	1.03	70.1	.4560	.3380	.6120	0.000	1.392	2.857
NASCAP 'YGOLDC'	.48	1.49	42.0	.3730	.2760	.6170	.810	1.392	2.444
NASCAP 'BUUNAT'	.59	1.86	63.4	.4380	.3250	.6130	1.443	2.596	4.203
NASCAP 'NL12'	.30	1.00	6.0	.0800	0.0000	0.0000	0.000	0.000	.602

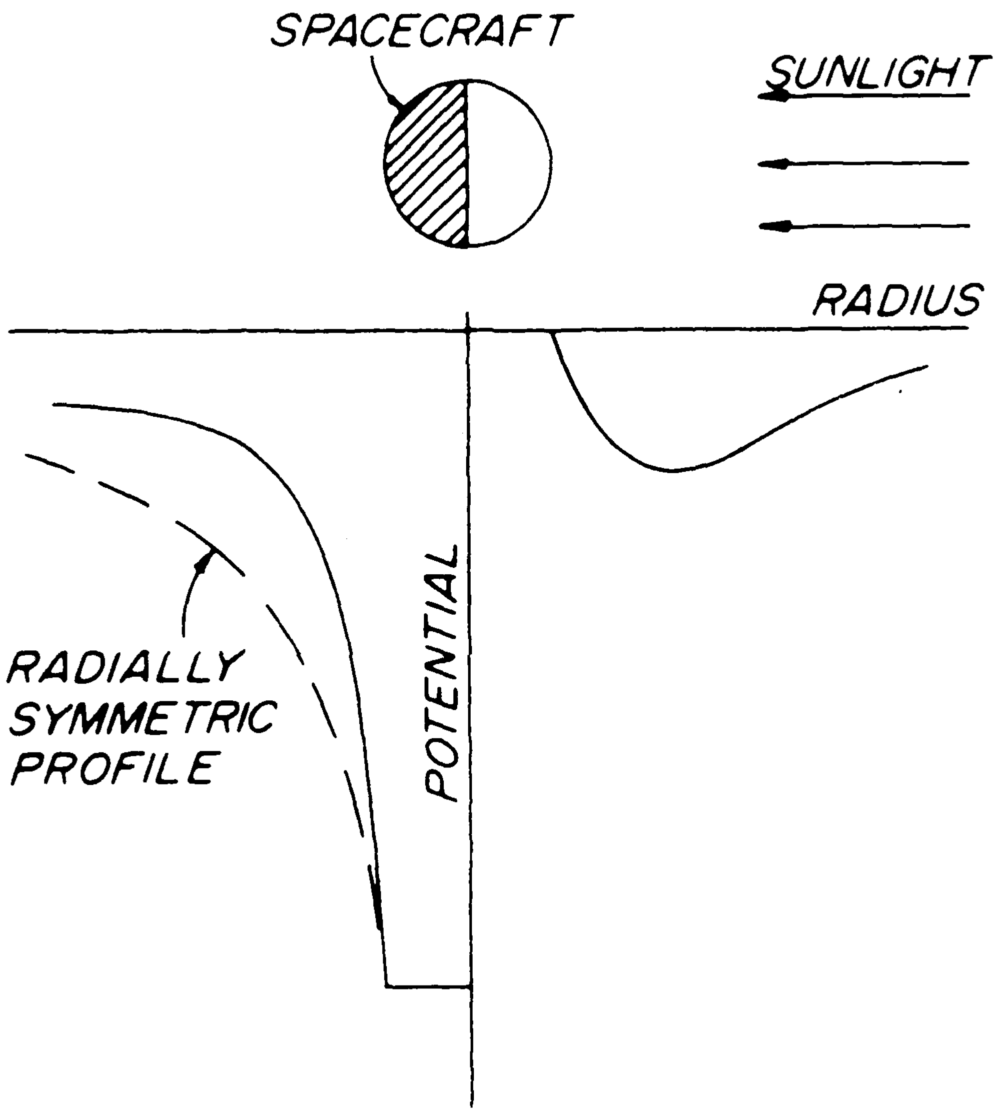


Fig. 2.1. Steepening of shaded-side potential profile, and sunlit-side potential barrier formation, caused by shaded-sunlit asymmetry on a spacecraft with an insulated surface, after Fahleson [1973].

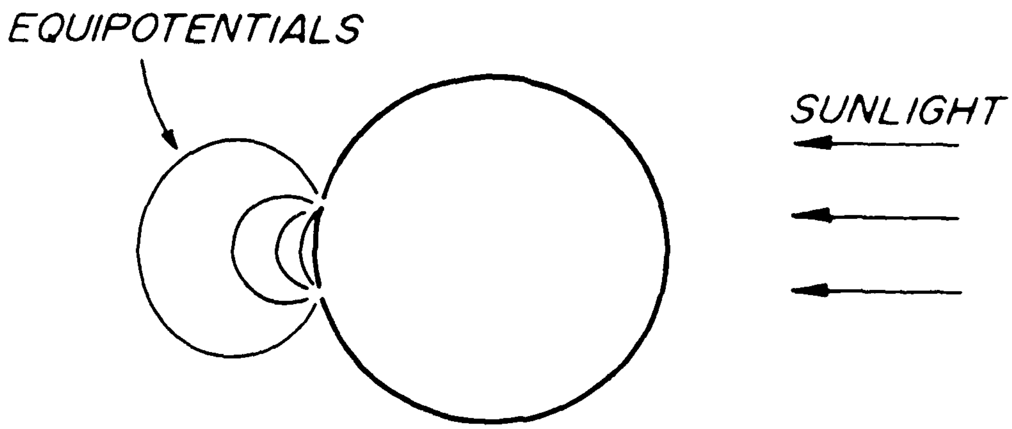


Fig.2.2. Conductive spacecraft with shaded isolated surface patch.

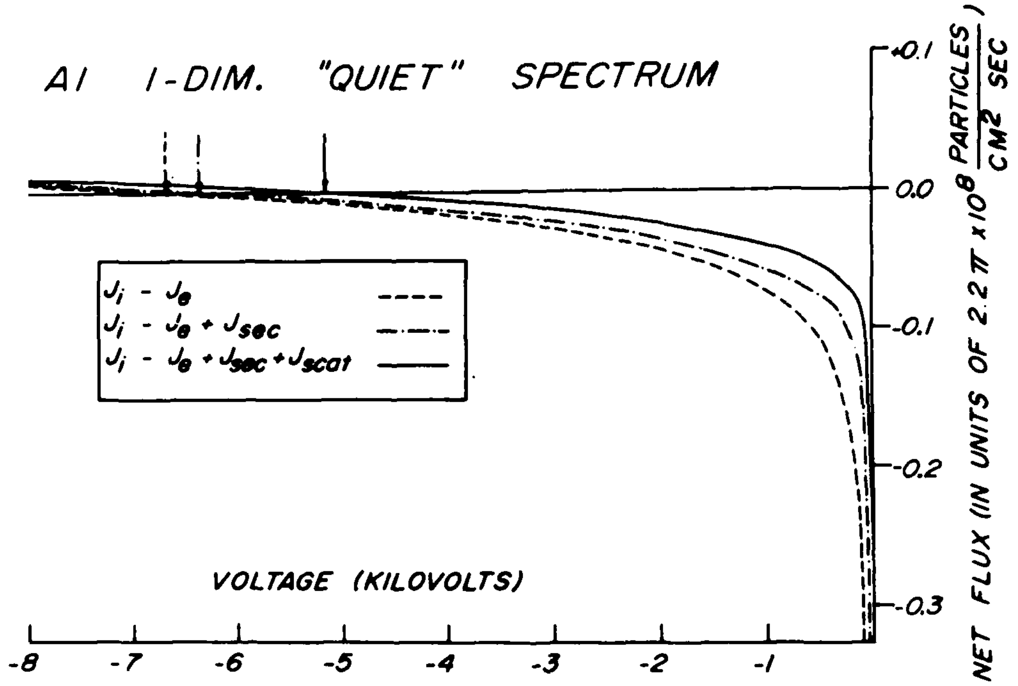


Fig. 2.3. Current-voltage characteristic for aluminum in "quiet" conditions, with a one-dimensional velocity-space cutoff. In Figures 2.3 - 2.7 the zeros of the characteristics are indicated by arrows.

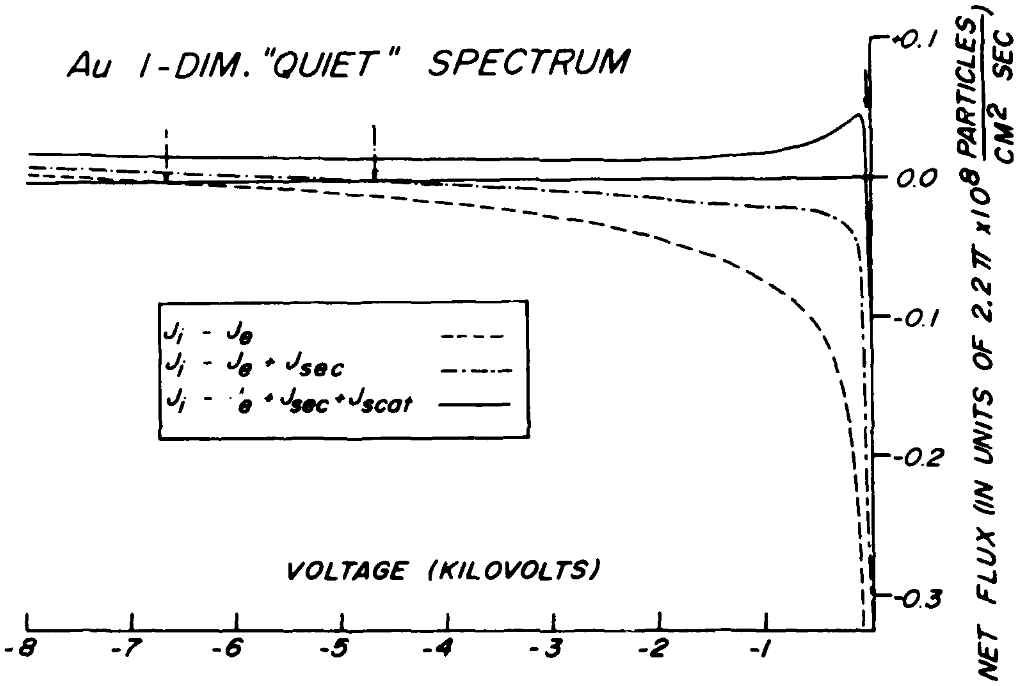


Fig. 2.4. Current-voltage characteristic for gold in "quiet" conditions, with a one-dimensional velocity-space cutoff.

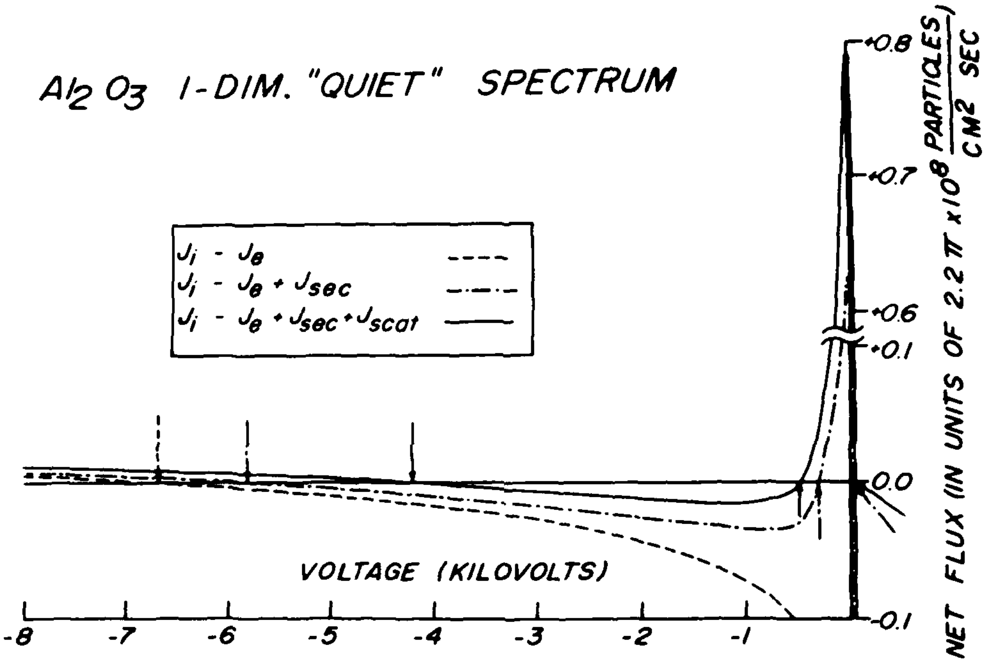


Fig. 2.5. Current-voltage characteristic for aluminum oxide in "quiet" conditions, with a one-dimensional velocity-space cutoff.

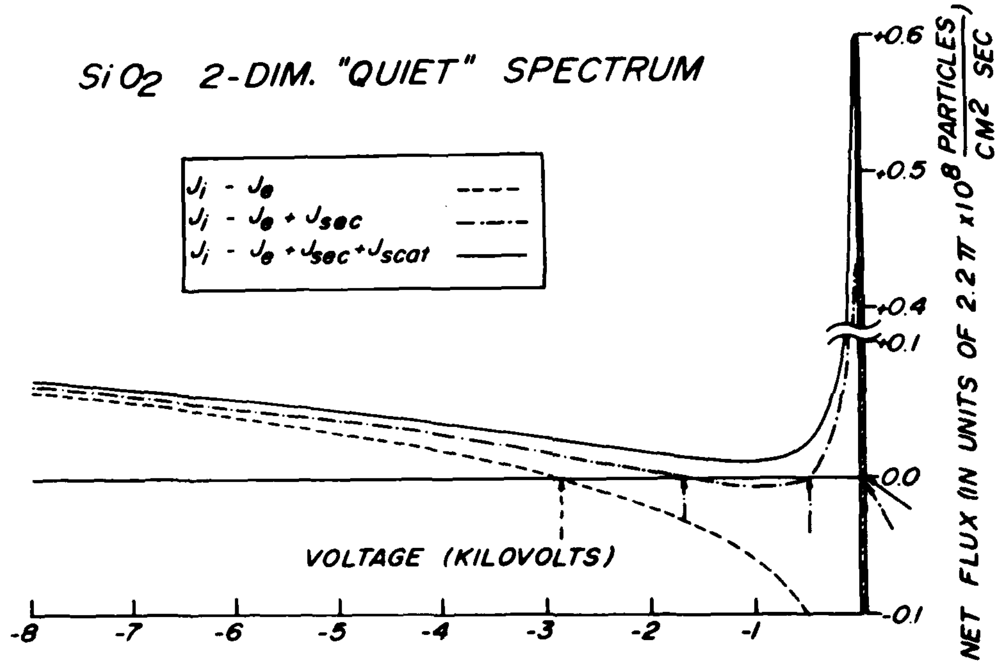


Fig. 2.6. Current-voltage characteristic for quartz in "quiet" conditions, with a two-dimensional velocity-space cutoff.

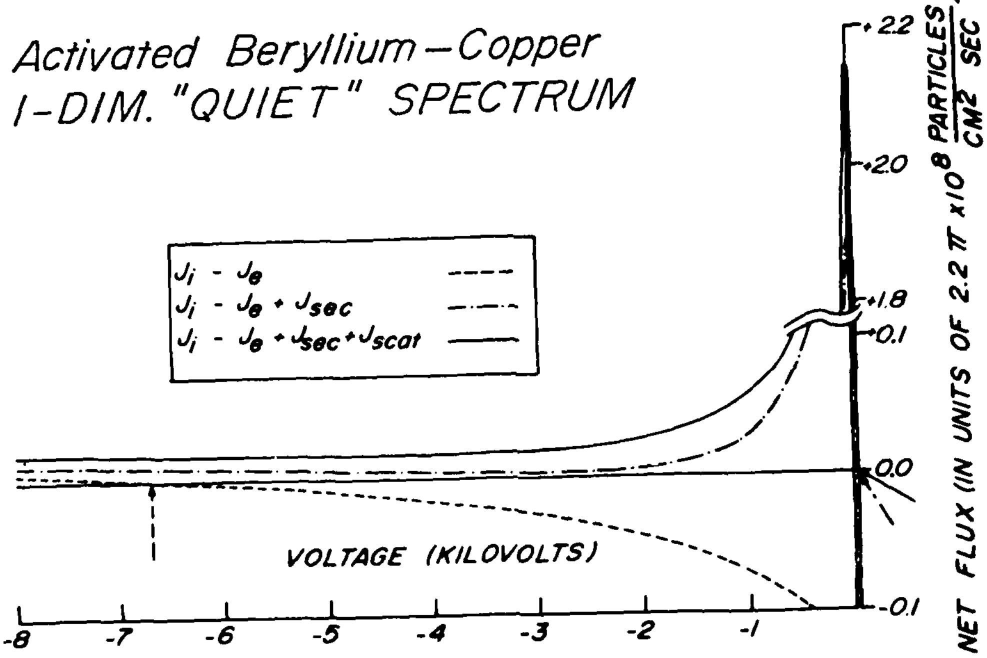


Fig. 2.7. Current-voltage characteristic for activated beryllium-copper in "quiet" conditions, with a one-dimensional velocity-space cutoff.

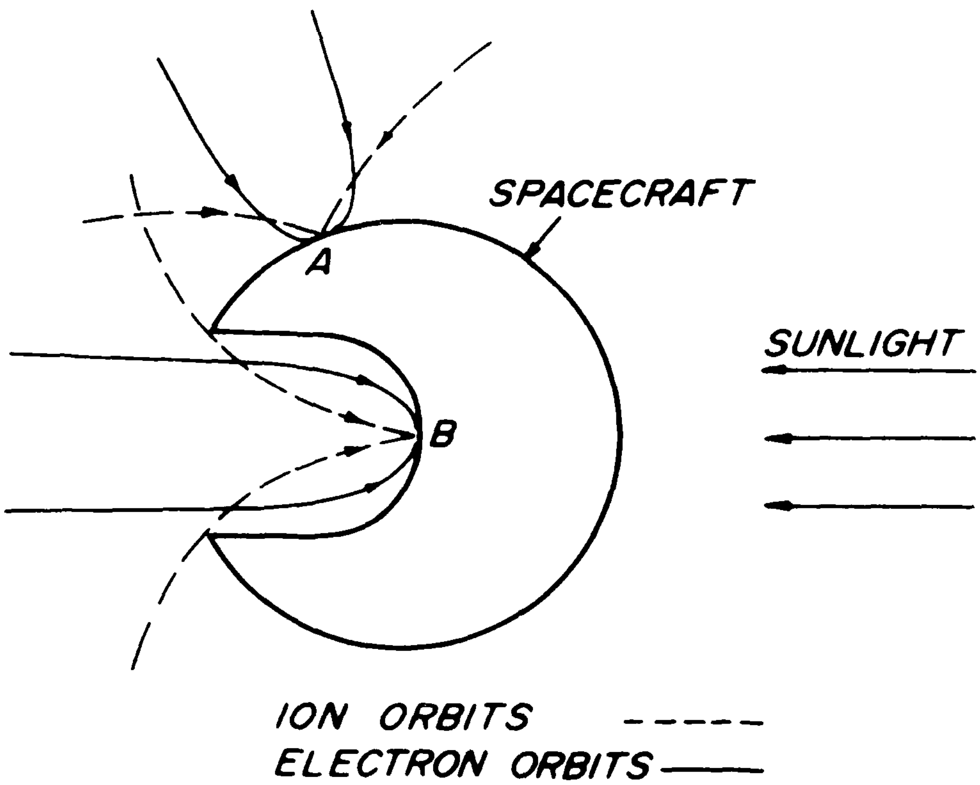


Fig. 2.8. Spacecraft with shaded isolated cavity, showing incident ion and electron orbits with energies close to the lowest for which collection of each species is possible. The orbits shown are cutoff orbits, defined as the most nearly tangential orbits for which incident particles of a given energy are able to reach a given point on the spacecraft surface, having tangential velocity component in a given direction.

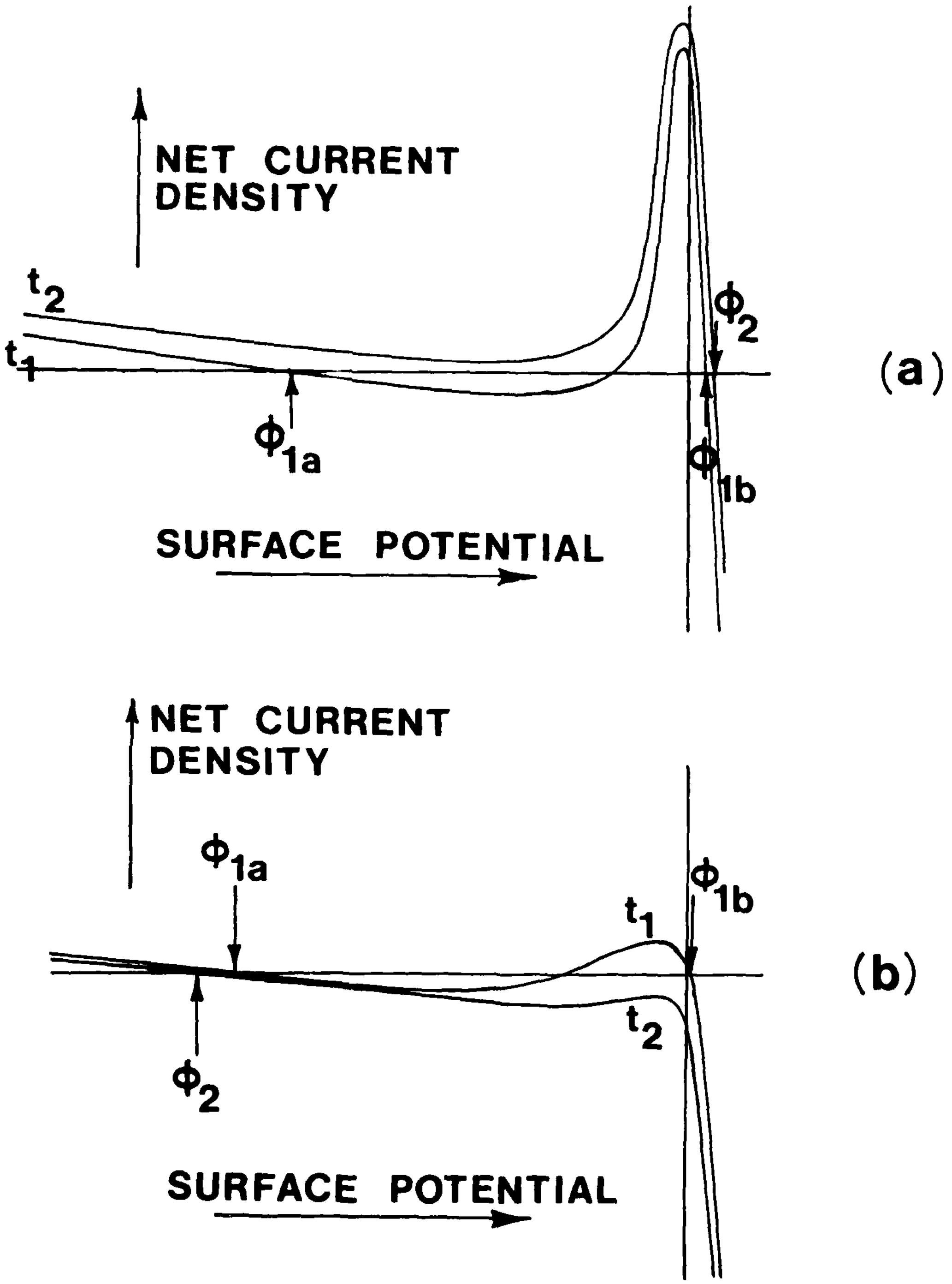


Fig. 2.9. Disappearance of multiple roots in the current-voltage characteristic of a spacecraft surface, as a result of temporal changes in ambient electron velocity distributions. These illustrations are schematic only but are representative of predicted behavior for various surface materials. Some combinations of materials and environments for which such behavior is predicted appear as clusters of three numbers in Table 1.

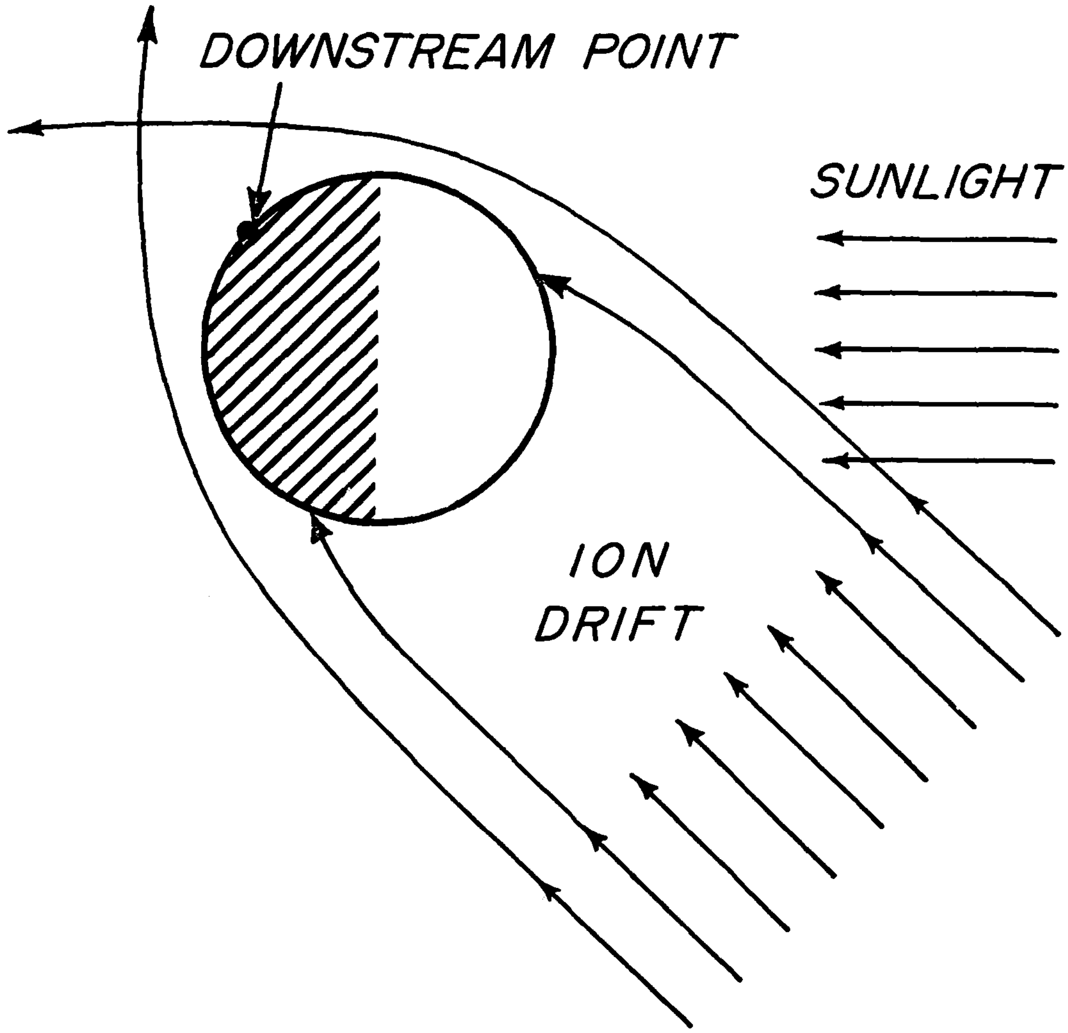


Fig. 2.10. Spherical spacecraft with downstream point (relative to ion drift direction) shaded.

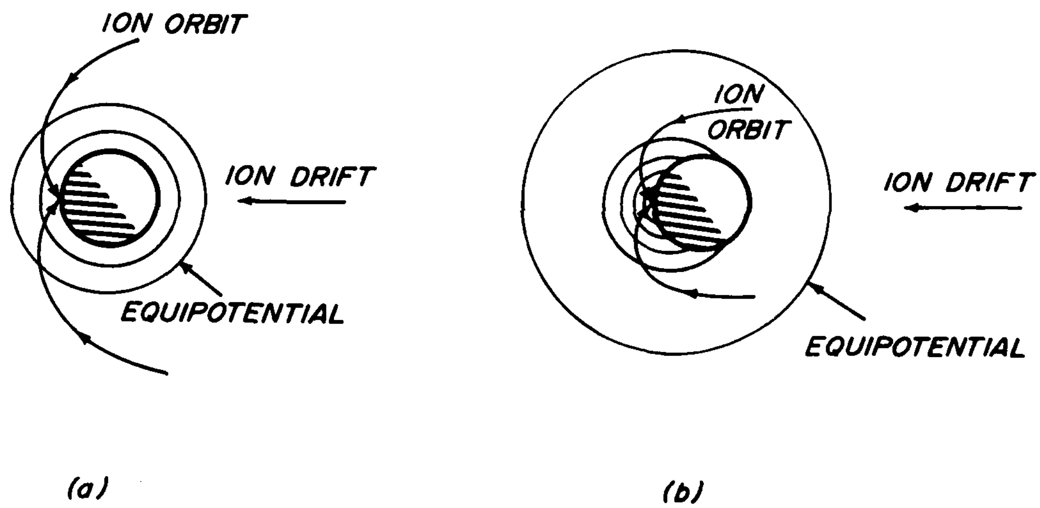


Fig. 2.11(a) Hypothetical symmetric equipotentials around a spherical spacecraft (b) nonsymmetric equipotentials around the same spacecraft.

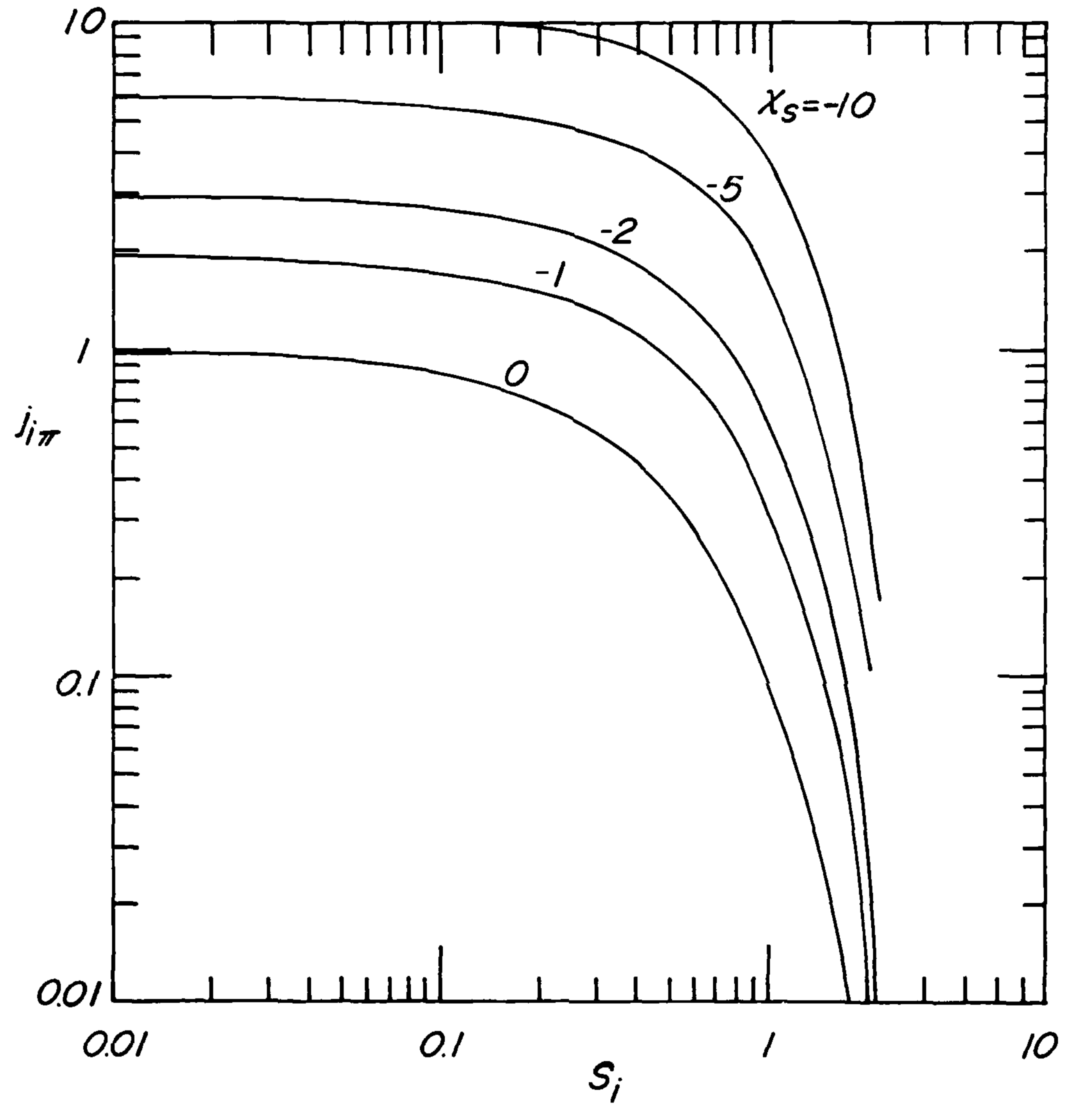


Fig. 2.12. Nondimensional downstream-point ion current density $j_{i\pi} = J_{i\pi}/[N_\infty e(kT_i/2\pi m_i)^{\frac{1}{2}}]$ as a function of ion speed ratio $S_i = U/(2kT_i/m_i)^{\frac{1}{2}}$ for various nondimensional surface potentials $\chi_s = e\phi_s/kT_i$, assuming spherical geometry, zero magnetic field, uniform surface potential, collisionless large-Debye-length conditions, and drifting Maxwellian ions. For $S_i \to 0$, $j_{i\pi} \to 1 + |\chi_s|$ when $\chi_s < 0$.

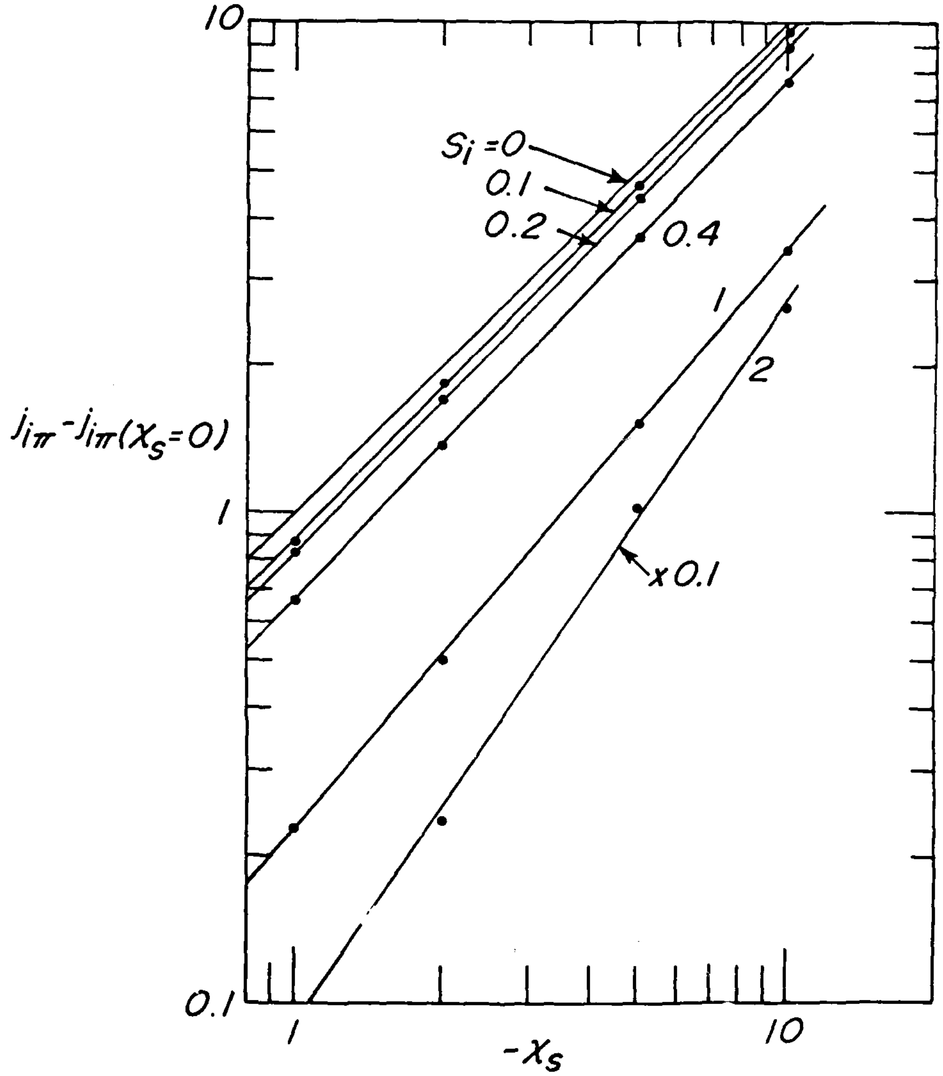
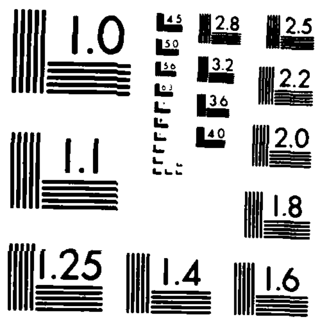


Fig. 2.13. Nondimensional downstream-point ion current density $j_{i\pi}$ as a function of surface potential χ_s for various ion speed ratios S_i, for the same conditions as in Fig. 2.12. The straight lines shown are power-law approximations.



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963 A

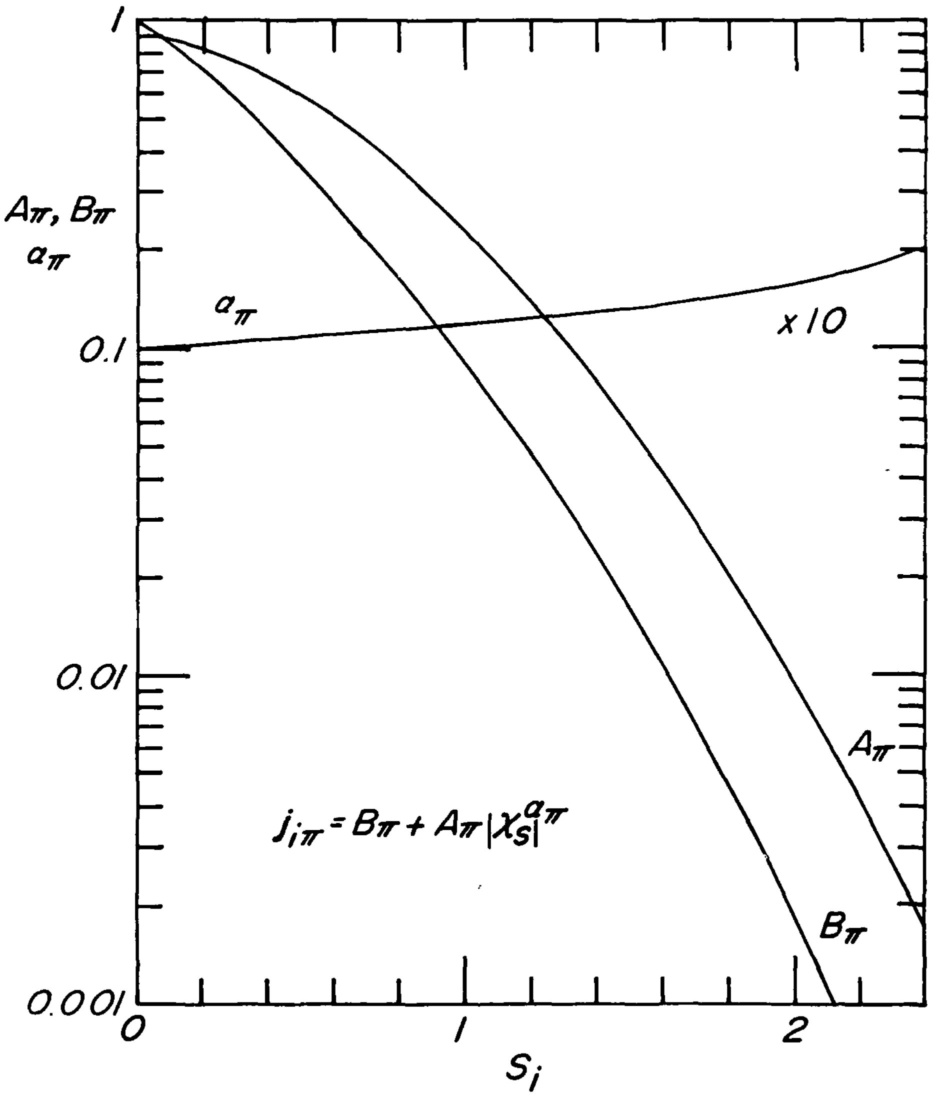


Fig. 2.14. Dependence of the power-law coefficients A_π, B_π and α_π on ion speed ratio S_i.

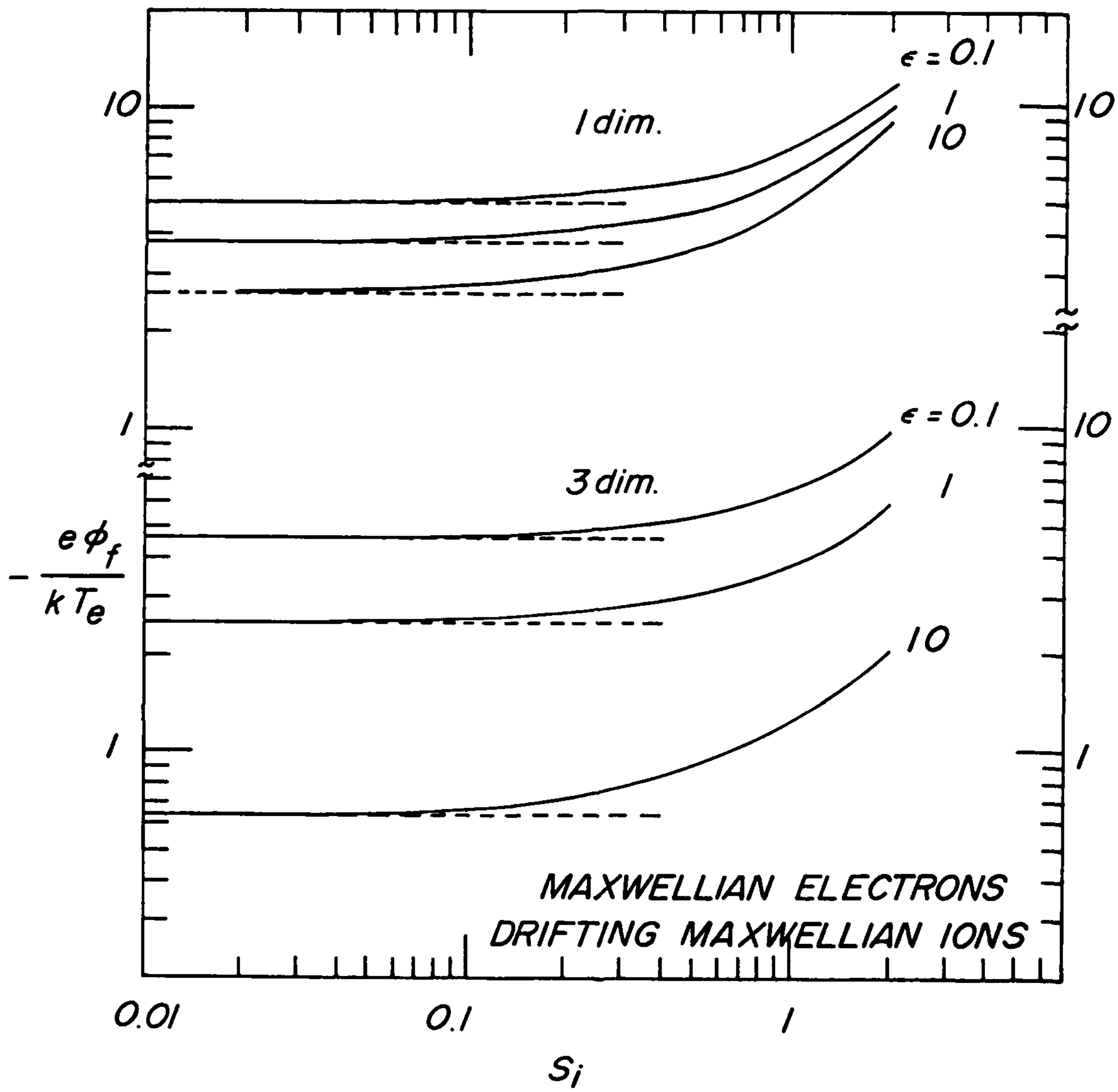


Fig. 2.15. Upper and lower bounds on floating potential ϕ_f at shaded downstream point of spacecraft, as a function of ion speed ratio S_i for various ion-to-electron temperature ratios ε, for Maxwellian electrons and drifting Maxwellian ions, for one-dimensional and three-dimensional ion velocity space cutoffs.

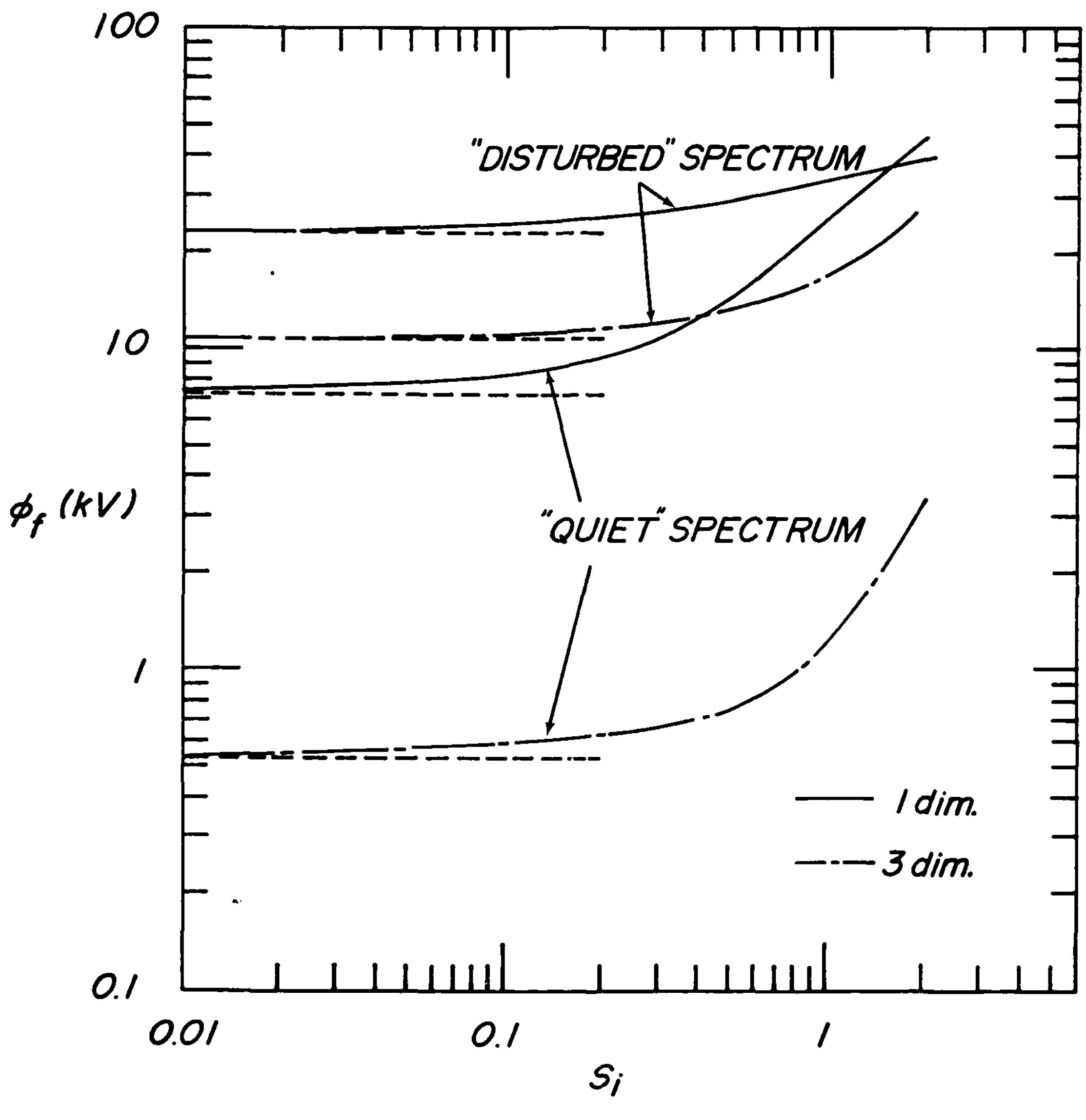


Fig. 2.16. Upper and lower bounds on floating potential ϕ_f at shaded downstream point of spacecraft, as a function of ion speed ratio S_i for "disturbed" and "quiet" electron velocity spectra representing geostationary orbit conditions, for one-dimensional and three-dimensional ion velocity space cutoffs.

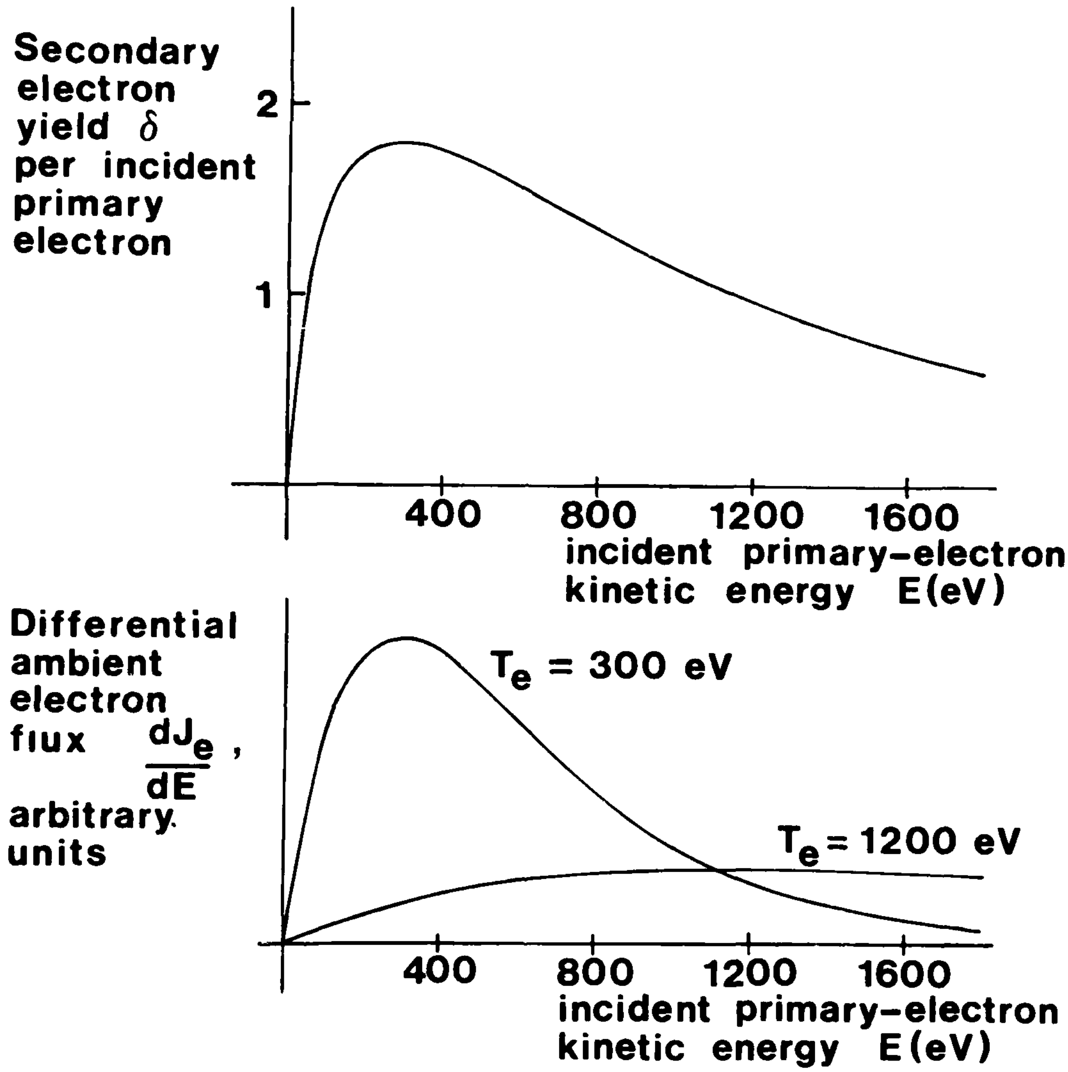


Fig. 3.1.(a) typical form of secondary-electron yield δ (secondary electrons per incident primary electron), as a function of incident primary kinetic energy E, at normal incidence (b) energy-differential electron flux dJ_e/dE for Maxwellian ambient electron velocity distributions at two different temperatures. Total secondary-electron flux is obtained by integrating the product of these two functions [multiplied by another factor given in Eq. (2.10)] over energy E.

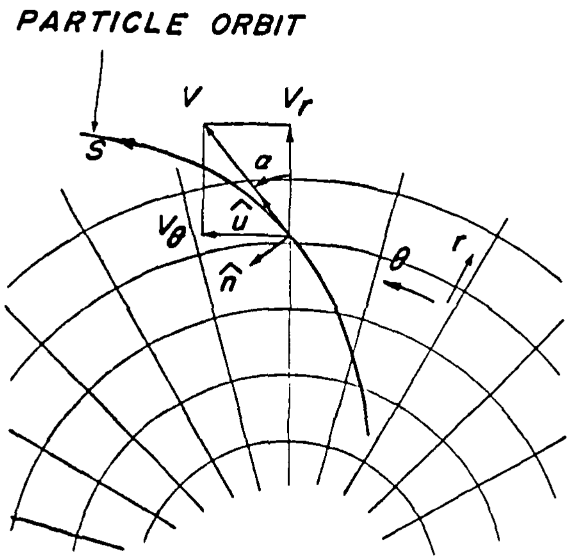


Fig. 4.1. Coordinate system and definitions for particle orbit integration.

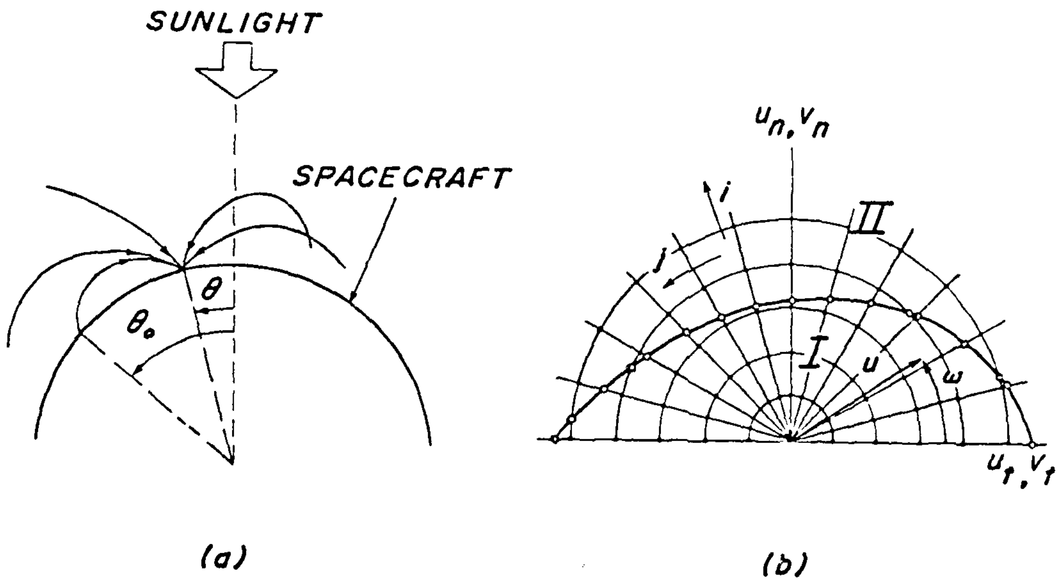


Fig. 4.2. Position-space (a) and velocity-space (b) coordinates for calculation of incident current density at a surface point. Figure 4.2a shows several particle orbits incident at a surface point θ, one of them having originated at the surface point θ_o.

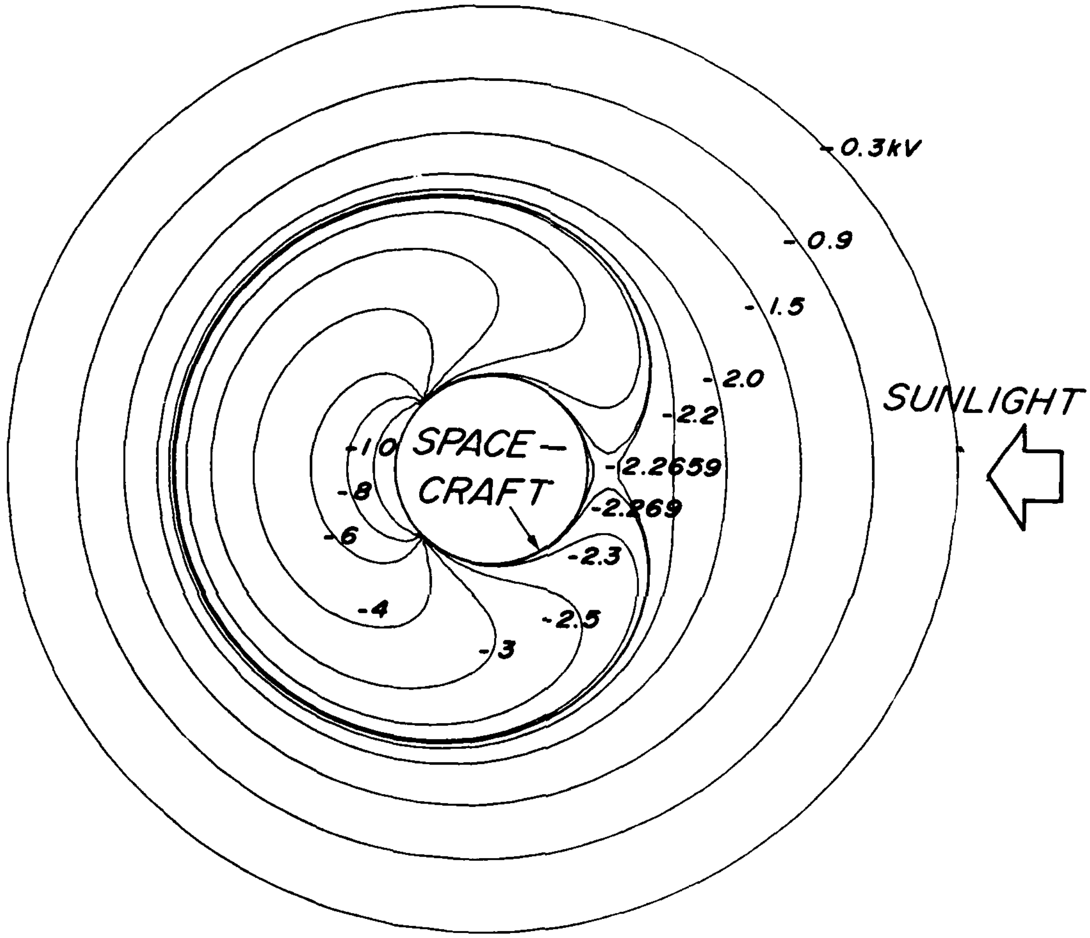


Fig. 4.3. Equipotential contours around a cylindrical spacecraft cross-section with two conductive sectors having angles of 270° and 90°. Sector potentials are -2.265 kV and -11.88 kV, respectively. Other data pertinent to this calculation are given in Sec. 4.5. The radial coordinate in Figs. 4.3 - 4.5, 4.8 and 4.9 is $1 + \ln(r/r_s)$ where r_s is spacecraft radius.

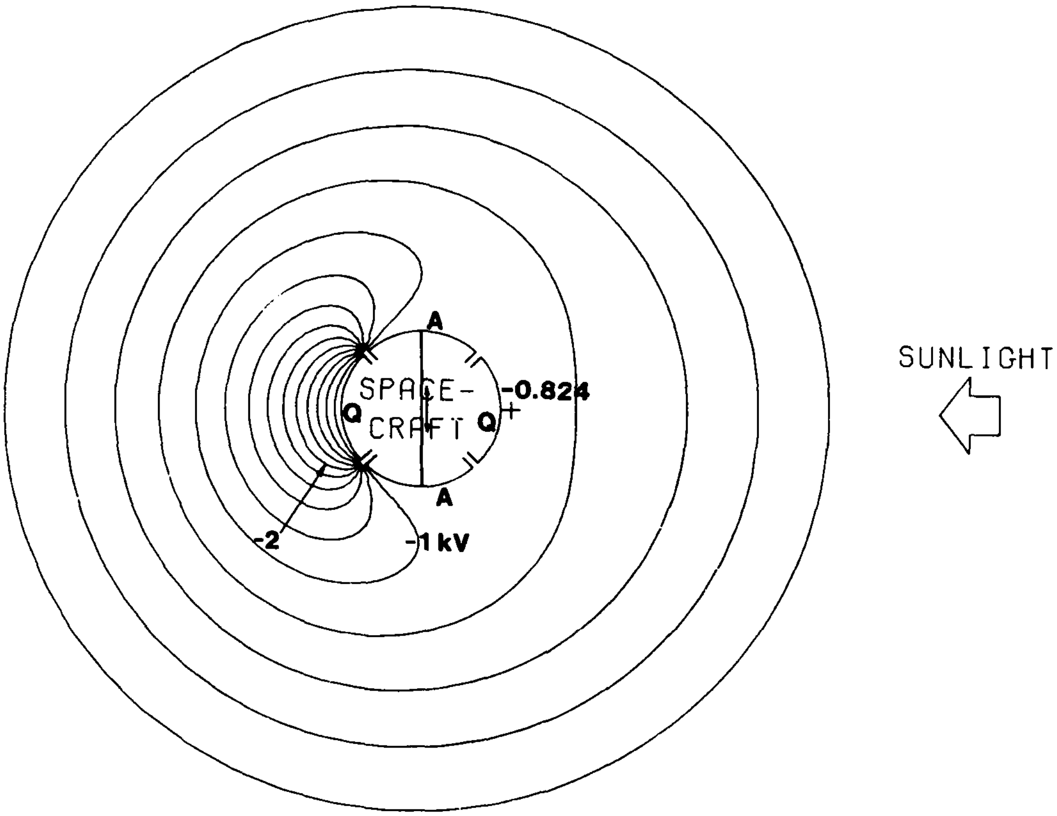


Fig. 4.4. CYLVIA calculation of equipotential contours around a cylindrical spacecraft cross-section with four sectors each having an angle of 90°. The two sectors labeled "A" are conductive, have the photoemission properties of aluminum, and are connected together electrically, as shown schematically in the figure. The two sectors labeled "Q" are nonconductive and have the photoemission properties of quartz. In this and subsequent figures, the symbol + indicates the location of a saddle point. Other data pertinent to this calculation are given in Sec. 4.5. The radial coordinate in this figure is as in Fig. 4.3. The floating surface potential distribution resulting from this calculation is shown in Fig. 4.6.

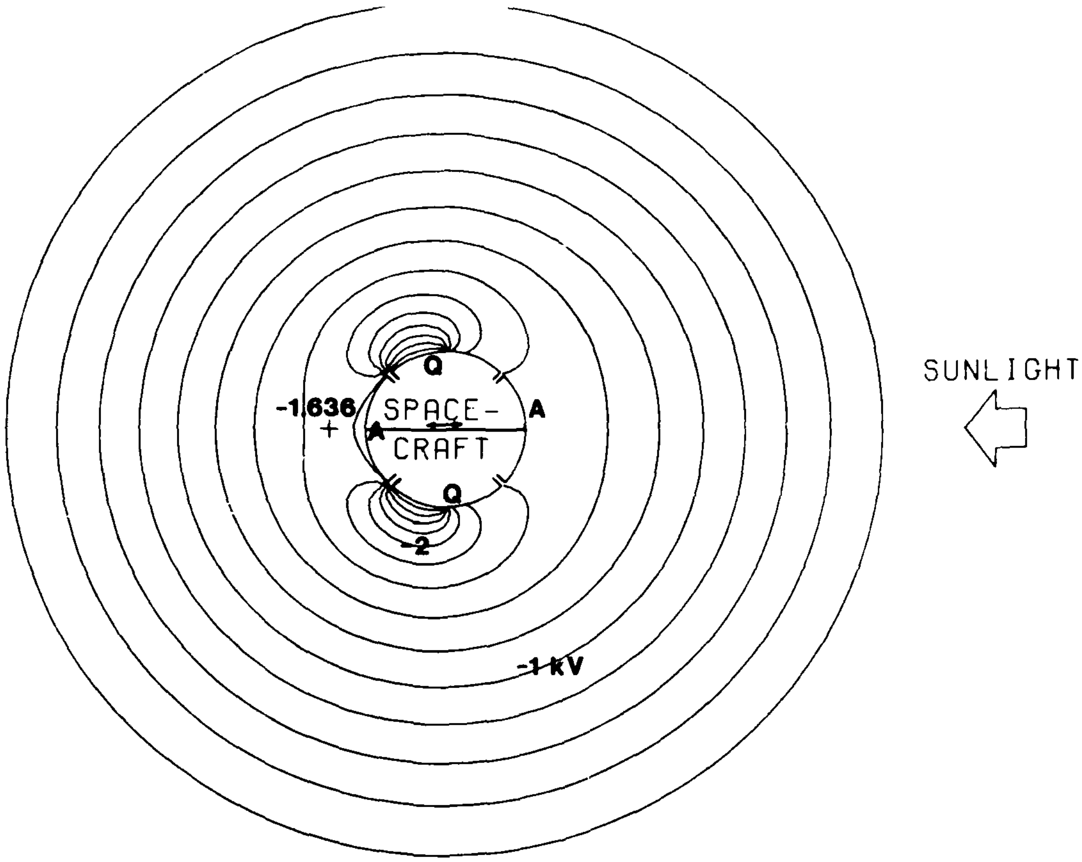


Fig. 4.5. Same as Fig. 4.4 except that the spacecraft has been rotated by 90°.

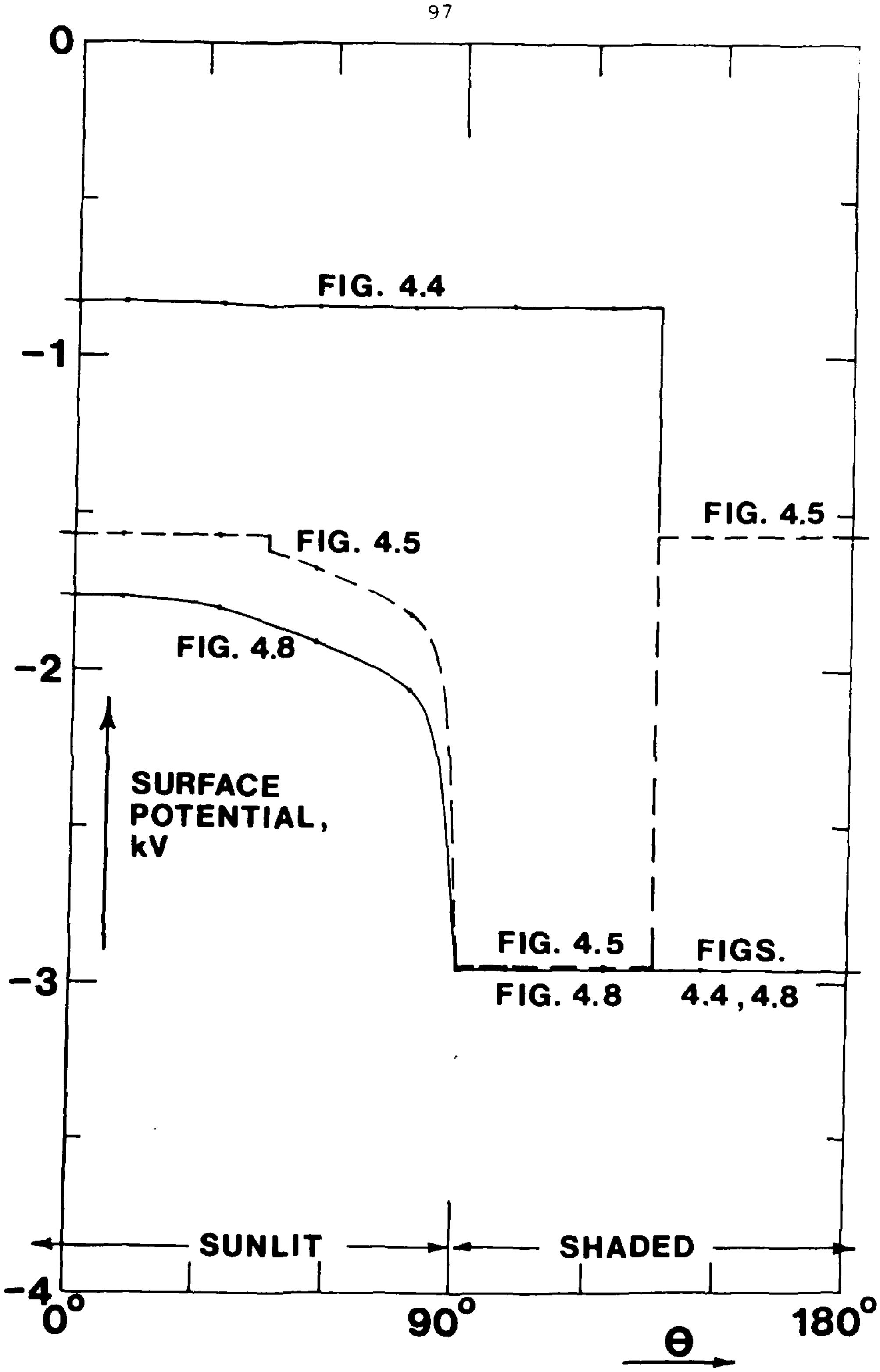


Fig. 4.6. Distributions of floating surface potential vs angle for the calculations shown in Figs. 4.4, 4.5, and 4.8.

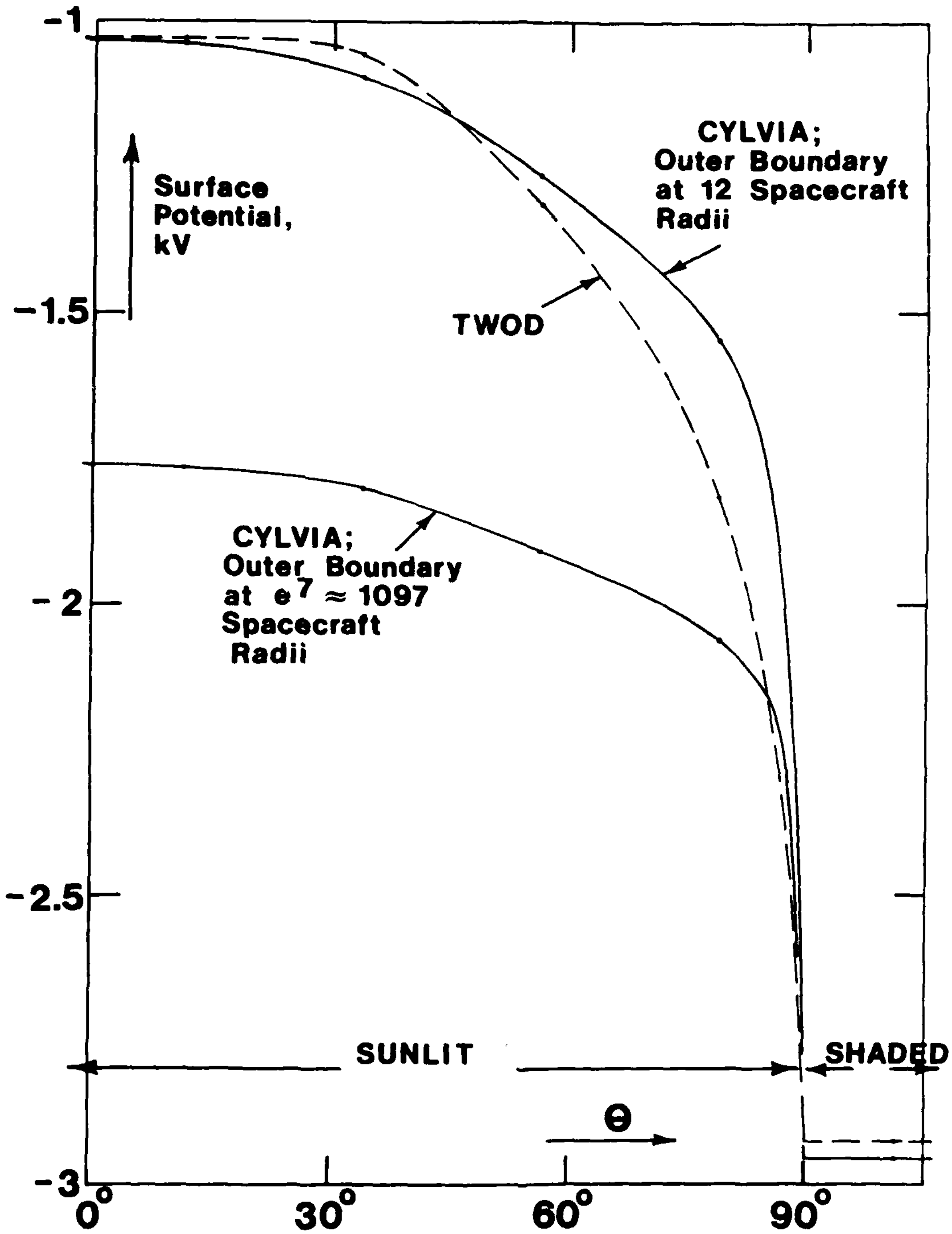


Fig. 4.7. Distributions of floating surface potentials vs angle for the calculation shown in Fig. 4.9, and also for another one which is the same except that the outer boundary of the calculation (which is held at space potential in all cases) is at $12r_s$ rather than e^7r_s. Also shown is another result for which the outer boundary is at $12r_s$, calculated using a program called TWOD, which combines NASCAP physical assumptions with circular cylindrical geometry (M. Mandell, Systems, Science and Software, Inc., private communication). Other data pertinent to these calculations are given in Sec. 4.5.

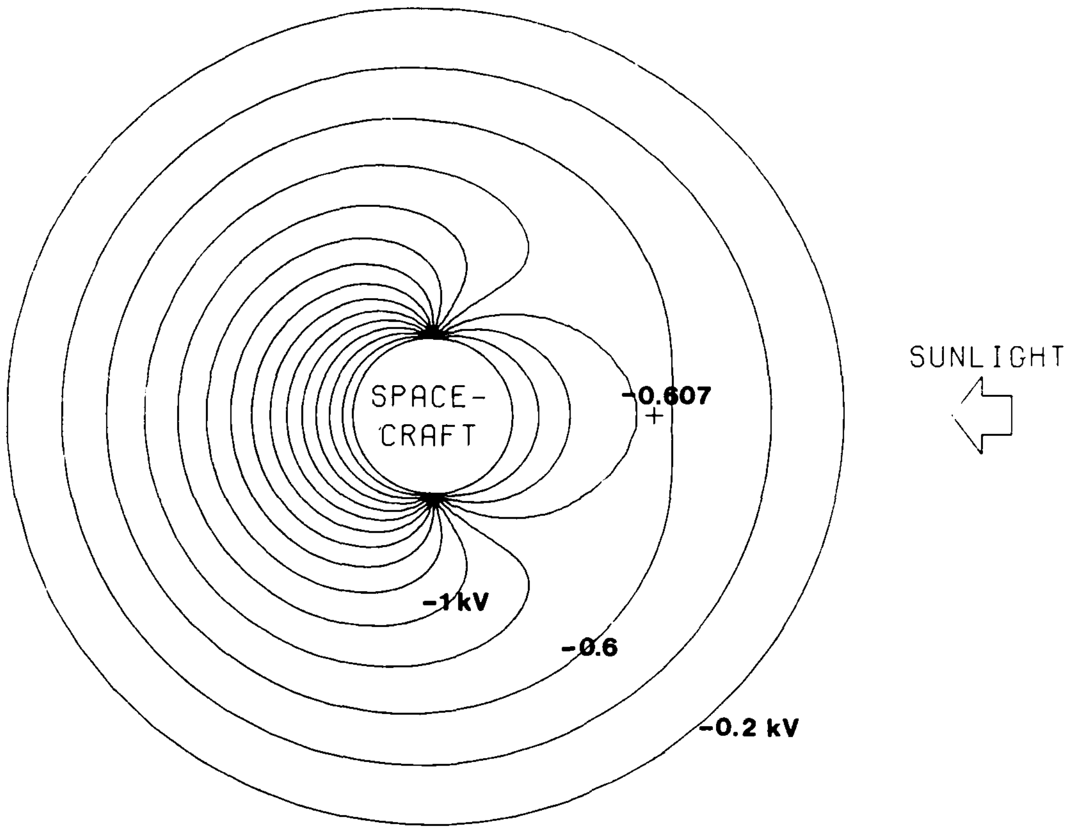


Fig. 4.8. CYLVIA calculation of equipotential contours around a cylindrical spacecraft cross-section having a nonconductive surface, corresponding to a surface potential distribution calculated assuming local current balance only (i.e., the presence of a potential barrier is ignored in calculating photoelectron escape). The symbol + indicates the location of a saddle point. The photoelectron charge flux $eJ_{ph} = 4.2\times10^{-5}$A/m^2 at normal incidence. $T_{ph} = 1.5$ eV. Ambient ion and electron distributions are single Maxwellians, each with $n_\infty = 3$cm^{-3} and $T = 1$ keV. Secondary and backscattered electron fluxes are assumed zero. The polar-coordinate computation grids in position (r,θ) and in velocity space contain 65×16 and 8×16 intervals, respectively. Outer grid boundary radius r_B is e^7 times spacecraft radius r_S. Linearized ambient space charge (Eq. 4.3; Laframboise and Prokopenko, 1977), corresponding to a Debye length of 96 r_S, is assumed. Radial coordinate in Figure is $1+\ln(r/r_S)$.

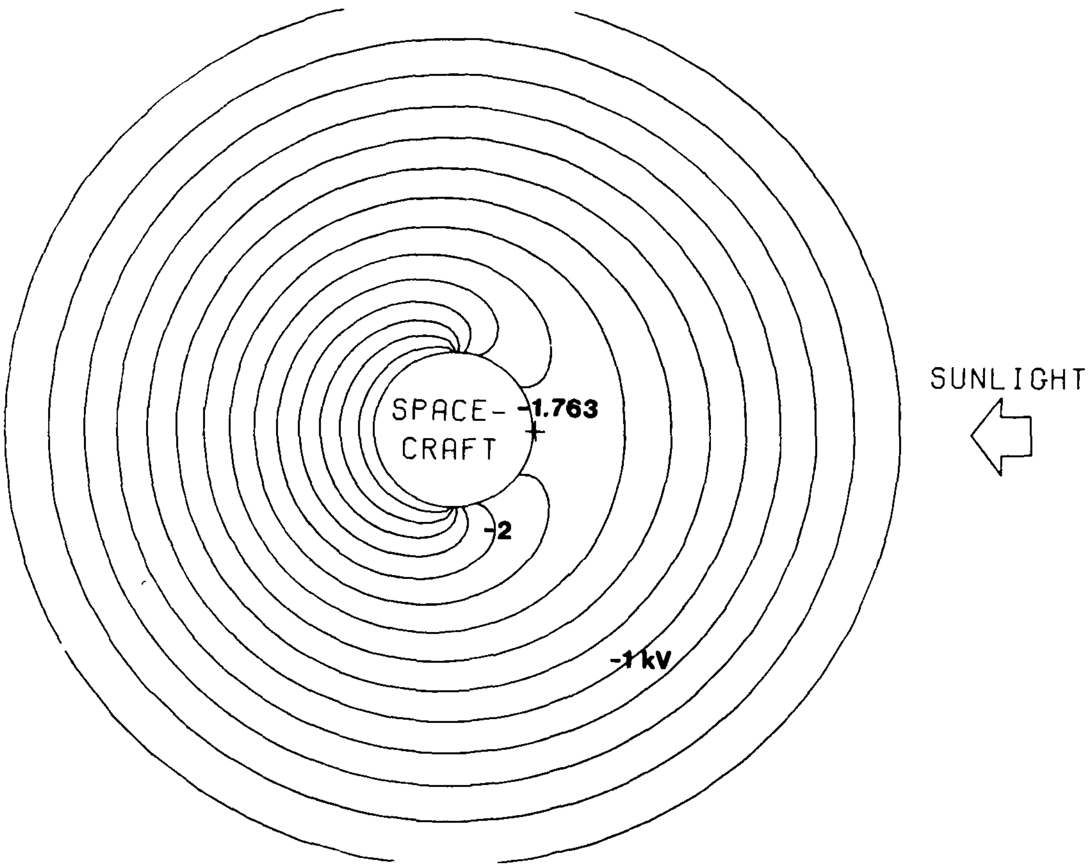


Fig. 4.9. Same as Figure 4.8 except that the calculation is now self-consistent including potential-barrier effects on photoelectron and other particle orbits.

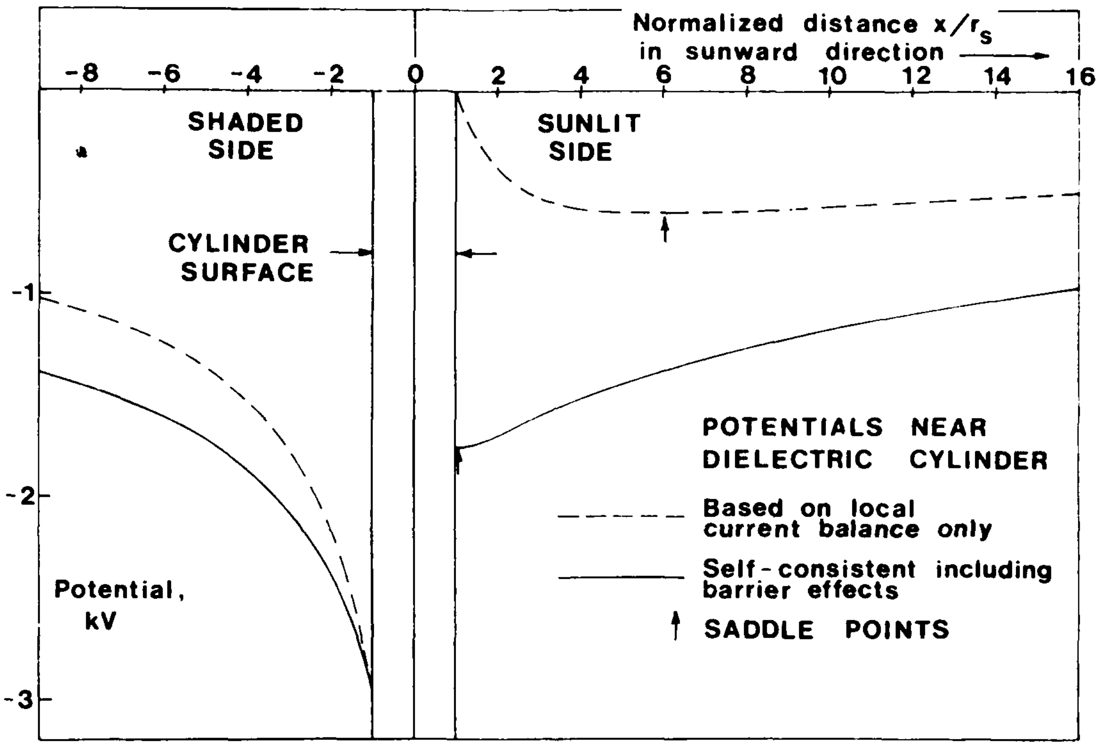


Fig. 4.10. Potentials vs distance along spacecraft-sun line for the situations shown in Figs. 4.8 and 4.9, showing the large decrease in differential charging which results when potential-barrier effects on particle (especially photoelectron) orbits are included. Without these effects, surface potentials at the sunward and anti-sunward points are 5.1V and -2.96 kV; with these effects, these potentials are -1.76 kV and -2.96 kV.

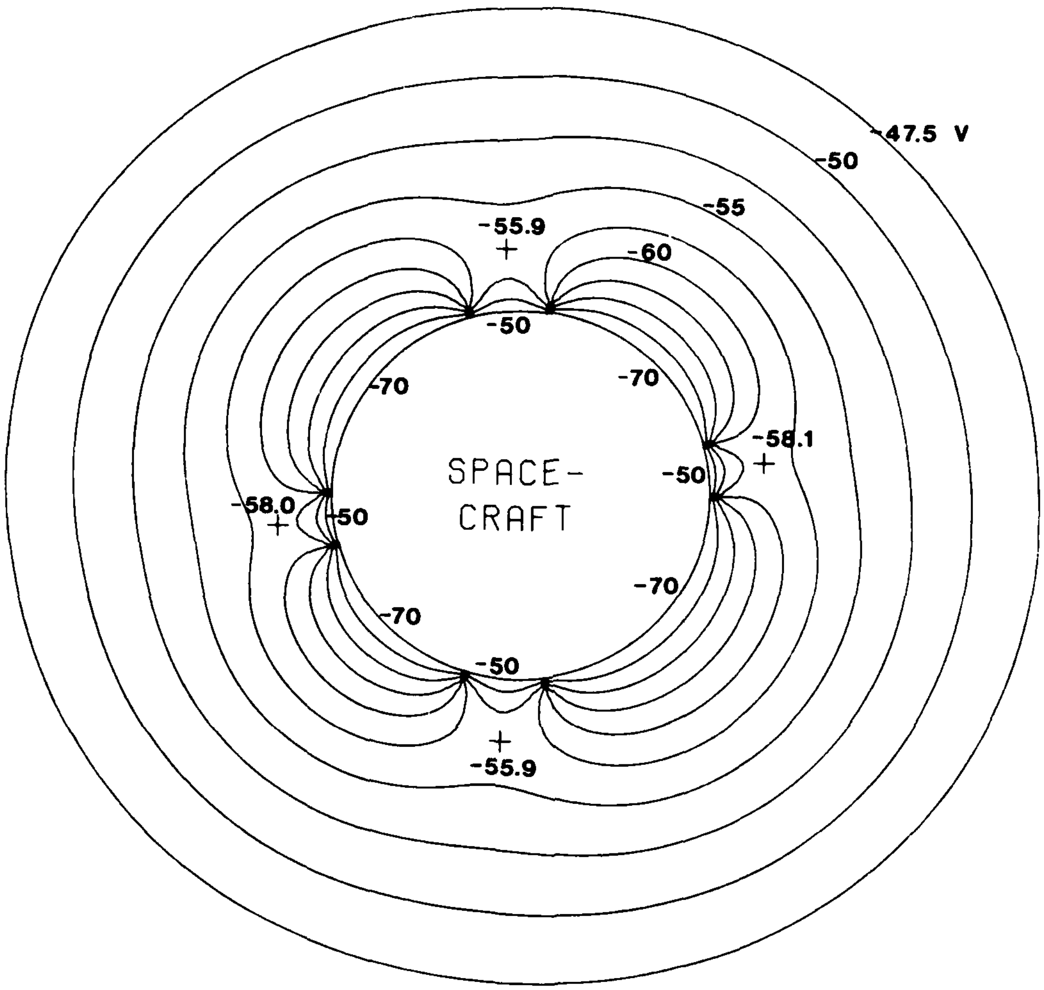


Fig. 4.11. CYLVIA calculation of equipotentials around a circular spacecraft cross-section with given rather than self-consistent surface potentials, as shown. In this Figure, the radial coordinate is proportional to radius. The computational grid in (r, θ) contains 65×180 intervals. The outer grid boundary is at $e^5 \approx 148$ spacecraft radii. Linear space charge corresponding to $\lambda_D = 32.5$ spacecraft radii is assumed. Other data pertinent to this calculation are given in Sec. 5.2. This calculation was done for the purpose of comparison with a XYCIC calculation shown in Fig. 5.1, and also with a similar but three-dimensional calculation done using NASCAP (Olsen, 1980, p. 190; Olsen and Whipple, 1980, Fig. 16). The results of this comparison are discussed in Sec. 5.2.

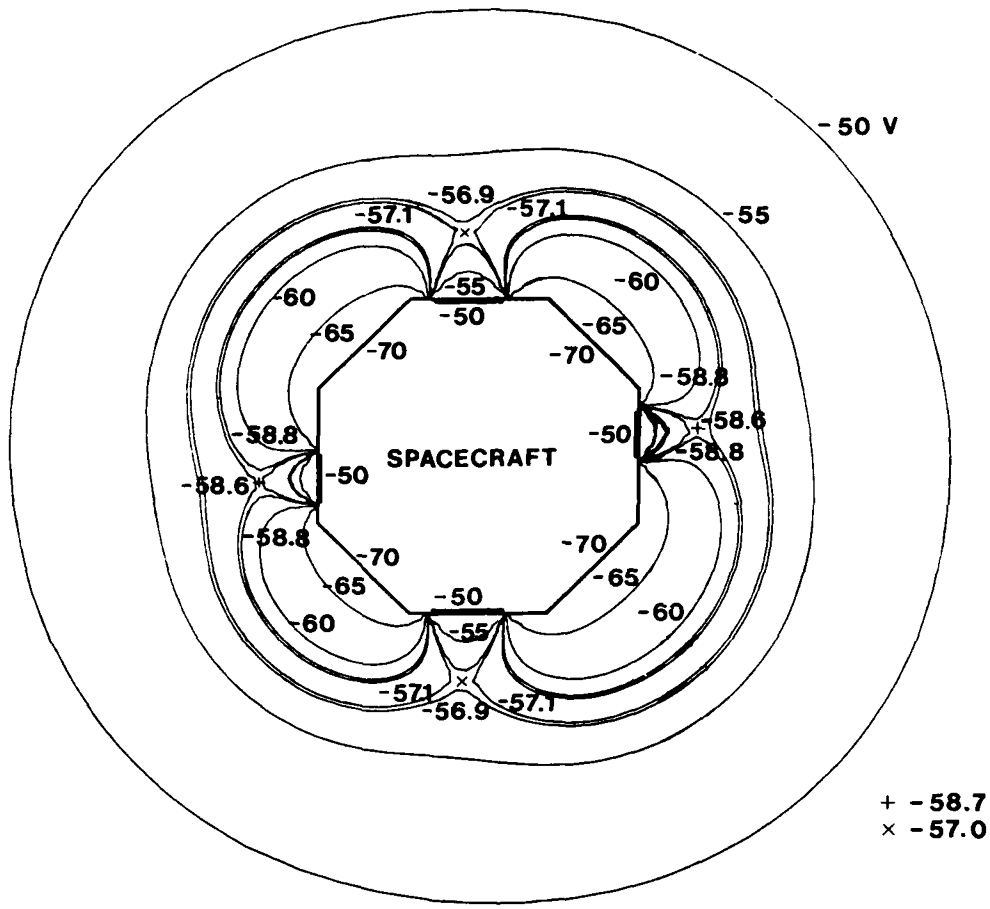


Fig. 5.1. XYCIC calculation of equipotentials, corresponding to the CYLVIA calculation shown in Fig. 4.11. One difference between these two calculations is that in the XYCIC calculation, zero space charge is assumed. The innermost grid used in the XYCIC calculation contains 40 × 40 intervals; the octagon has a dimension of 28 × 28 intervals and is centred in this grid. Six other grids surround the innermost grid; the outermost grid boundary is a centred square of side 19.2 times as large as the innermost grid boundary, i.e. located at 27.4 object half-widths.

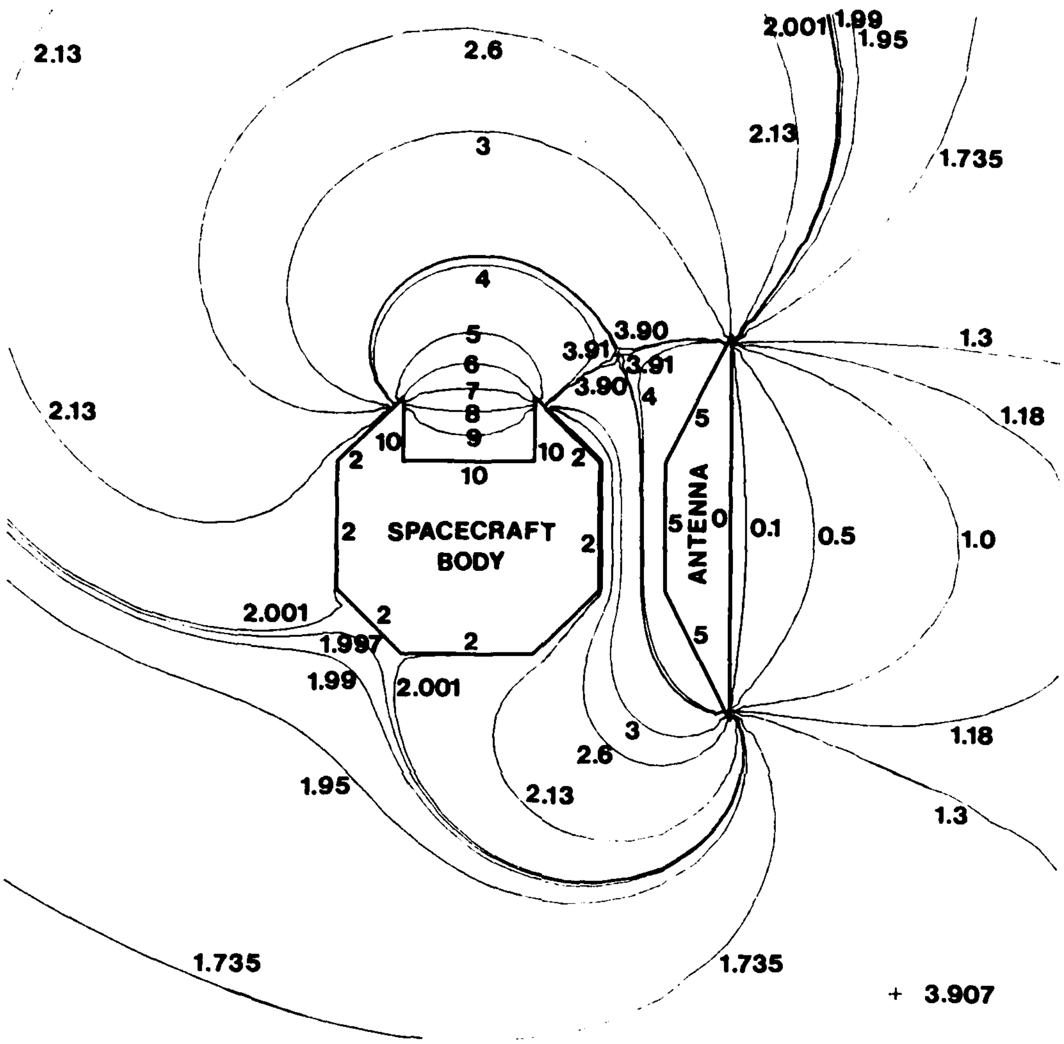


Fig. 5.2. XYCIC calculation of equipotentials around a cross-section through a hypothetical antenna + spacecraft body combination. As in Figs. 4.11 and 5.1, surface potentials are imposed (in this case, also hypothetical) values. Zero space charge is assumed. The innermost grid used contains 32×32 intervals with the combined total width of the "object" being 24 intervals. Six other grids surround the innermost grid; the outermost grid boundary is a centred square of side 20 times the innermost grid boundary, i.e. located at 26.7 half-widths of the combined object.

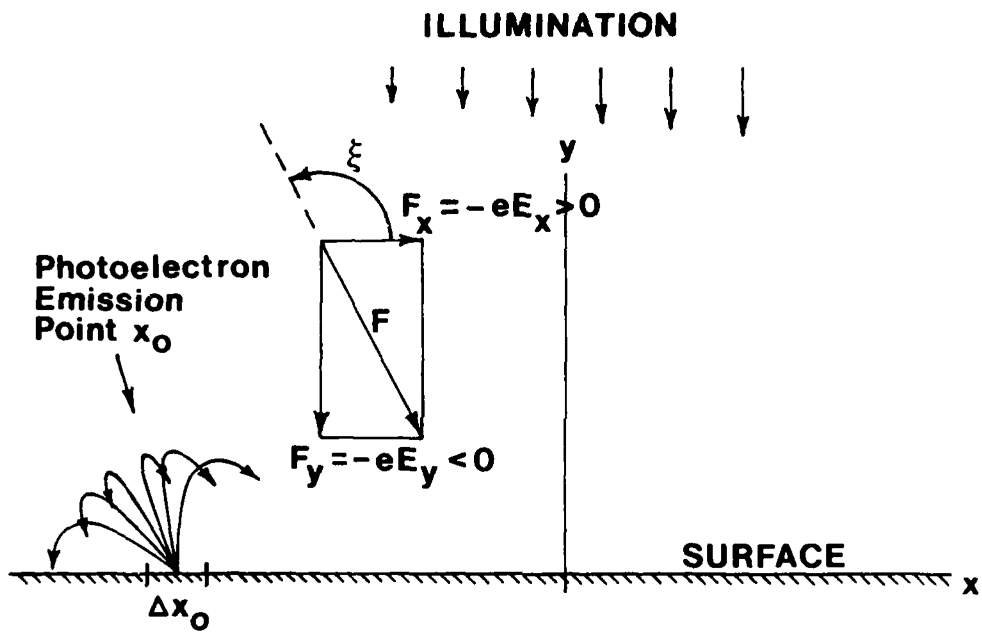


Fig. 6.1. Geometry for photocurrent sheet calculation. The illumination is assumed to vary linearly with x, producing a photoemission current density $J_{ph}(x_o) = J_{ph,o} + J_{ph}'x_o$. The electric field components E_x and E_y are assumed <u>uniform</u>, so that the surface potential varies linearly with x. The electron orbits are then tilted parabolas. Their impact points x can be found analytically (Eq. 6.1) for given values of emission position x_o and emission velocity components v_{xo} and v_{yo}.

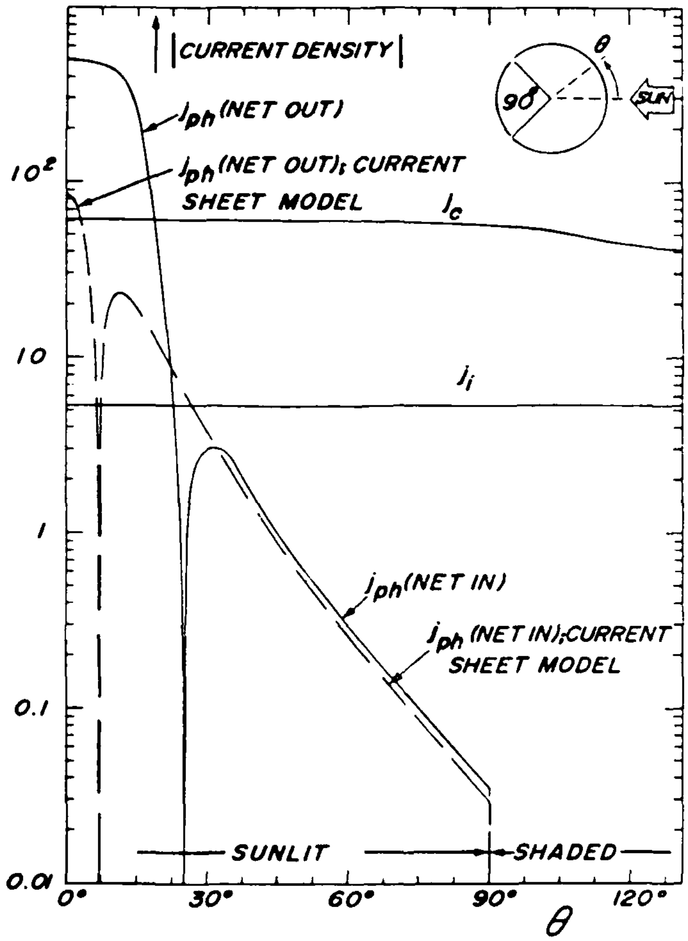


Fig. 6.2. Current densities vs surface position for the situation shown in Fig. 4.3 and described in Sec. 6.3. In this Figure, normalized current density j is defined as J/J_{ref}, where J_{ref} is the random flux of Maxwellian ions having a temperature and density of 1 keV and 1 cm^{-3}. Approximate photoelectron currents j_{ph} obtained using the approximate surface current ("current sheet") model given by Eqs. (6.7) and (6.8) are shown as dashed curves.

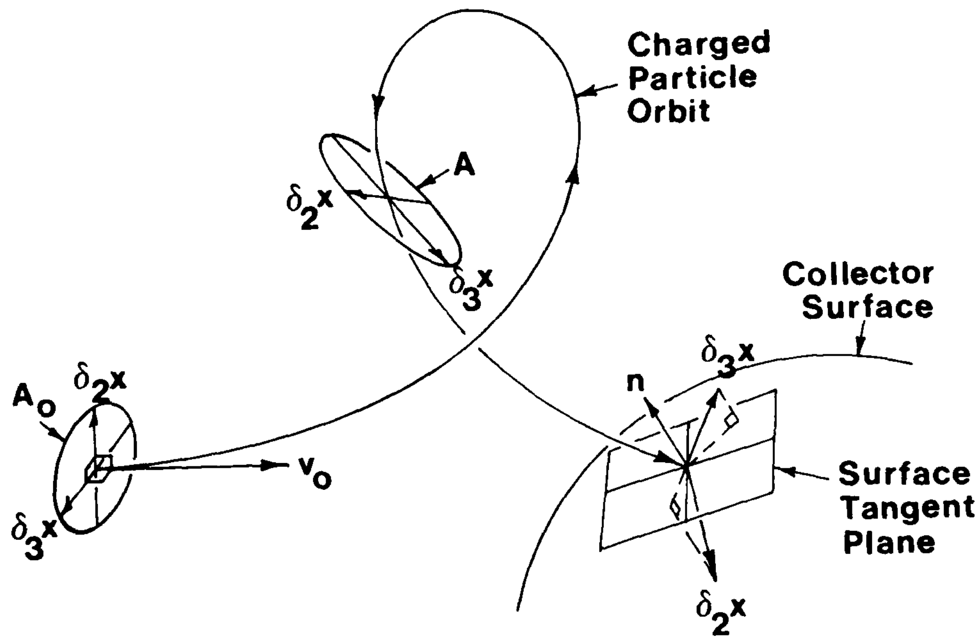


Fig. 7.1. Particle orbit geometry.

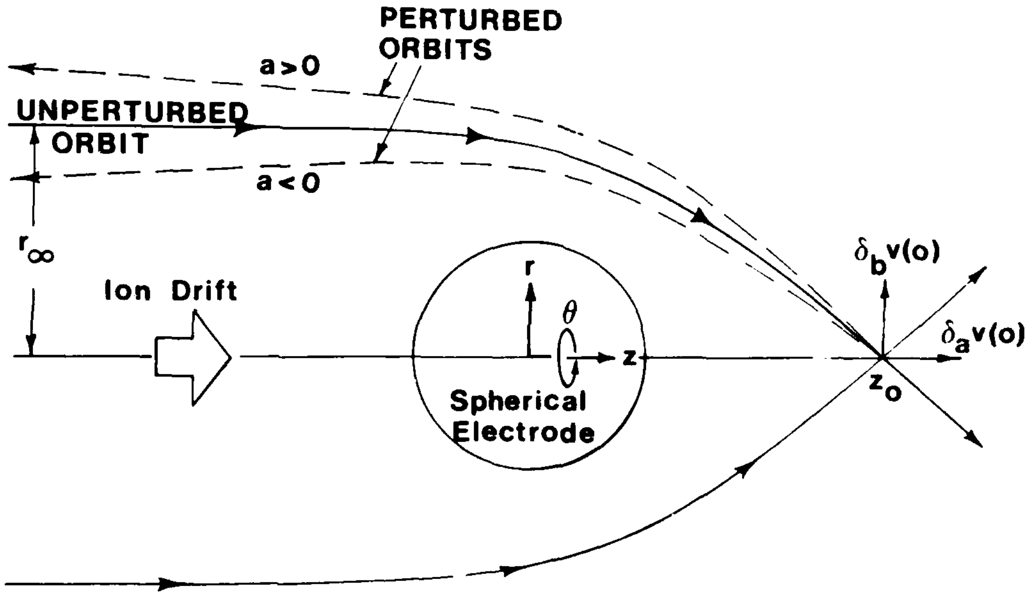


Fig. 7.2. Geometry for perturbation calculation of sphere axial defocusing.

Appendix A: Listing of LOCHG

```
      COMMON /RSPEED/SPEED,ASPEED,BSPEED,CSPEED
      COMMON /SCAT/WEIGHT(5),SCATA(5),SCATB(5),SCATC(5),VSCAT(5),NUMPAR
      COMMON /GENRAL/EMAX,DELMAX,TSEC,RION,TION,RPI,CUTOFF,CUTLOW,KBRNCH
      COMMON /Z/H(100),Y(100),KI(5),M
      COMMON/NV/N,V,K
      READ(14,*)N,M,K
      READ(14,*)(Y(I),H(I),I=1,M)
      RPI=1.128379
      READ(14,*)RION,TION,TSEC
      WRITE(14)RION,TION,TSEC
   24 READ(14,*)(KI(I),I=1,K)
      KBRNCH=KI(1)
   22 READ(5,11)V,CUTOFF,CUTLOW
      WRITE(14)CUTOFF,CUTLOW
   15 READ(14,*)SPEED,ASPEED,BSPEED,CSPEED
      WRITE(14,102)SPEED,ASPEED,BSPEED,CSPEED
  102 FORMAT(1X,'SPEED RATIO IS ',F10.5,'  SPEED RATIO CONSTANTS ARE',
     1RE'3E15.5)
    5 READ(14,*)DELMAX,EMAX,NUMPAR
      DO 16 I=1,NUMPAR
      IF(DELMAX.EQ.0.)GO TO 13
      READ(14,*)WEIGHT(I),SCATA(I),SCATB(I),SCATC(I),VSCAT(I)
   16 CONTINUE
      V1=V
      CALL CNET(V1,SUM1,KI,K,CION,CELEC,CSEC,CSCAT,M,H,Y)
      V2=V1
    6 V2=V2+20.*SUM1
      CALL CNET(V2,SUM2,KI,K,CION,CELEC,CSEC,CSCAT,M,H,Y)
      IF(SUM1*SUM2.GE.0.)GO TO 6
    4 V=(V1+V2)/2.
      CALL CNET(V,SUMNEW,KI,K,CION,CELEC,CSEC,CSCAT,M,H,Y)
      IF(V.EQ.V1.OR.V.EQ.V2)GO TO 7
      IF(SUM1*SUMNEW.LT.0.)GO TO 3
      V1=V
      SUM1=SUMNEW
      GO TO 4
    3 V2=V
      SUM2=SUMNEW
      GO TO 4
    7 IF(V.GT.0.)WRITE(14,100)V
  100 FORMAT('V IS GREATER THAN'E15.7)
   14 WRITE(14,8)(KI(I),I=1,4),EMAX,DELMAX,V,CELEC,CION,CSEC,CSCAT
    8 FORMAT(1X,4I2,2F6.3,5E15.7)
      WRITE(14,200)V,SUM1
  200 FORMAT('THE NET CURRENT AT'E15.7,'IS'E15.7)
      GO TO 5
   13 STOP
      END
      BLOCK DATA
      COMMON/RSPED/SPEED,ASPEED,BSPEED,CSPEED
      COMMON/GENRL/EMAX,DELMAX,TSEC,RION,TION,RPI,CUTOFF,CUTLOW,KBRNCH
      COMMON/Z/H(100),Y(100),KI(5),M
      COMMON/NV/N,V,K
      DATA N,M,K/14,14,4/
      END
      SUBROUTINE CNET(V,SUMNEW,KI,K,CION,CELEC,CSEC,CSCAT,M,H,Y)
      REAL KON,KON1,KON2,KON3,KON4,KON5,KON6,KON7,KON10,KON20,KON30,
     1KON40,KON50,KON60,KON70,KON11,KON21,KON31,KON41,KON51,KON61,KON71,
     1KON12,KON22,KON32,KON42,KON52,KON62,KON72
     1,KON13,KON23,KON33,KON43,KON53,KON63,KON73,KON8,KON80,KON81,KON
     182,KON83
      DIMENSION H(M),Y(M),KI(K)
      COMMON /GENRAL/EMAX,DELMAX,TSEC,RION,TION,RPI,CUTOFF,CUTLOW,KBRNCH
      IF(KI(1).EQ.0)CION=0.
      IF(KI(1).EQ.1)CALL ION1(V,CION)
      IF(KI(1).EQ.2)CALL ION2(V,CION)
      IF(KI(1).EQ.3)CALL ION3(V,CION)
      IF(KI(2).EQ.0)CELEC=0.
      IF(KI(2).EQ.1)CALL ELEC1(V,CELEC)
      IF(KI(3).EQ.0)CSEC=0.
      IF(KI(3).EQ.1)CALL SEC1(V,CSEC,M,H,Y)
      IF(KI(4).EQ.0)CSCAT=0.
      IF(KI(4).EQ.1)CALL SCAT1(V,CSCAT,M,H,Y)
      SUMNEW=CION+CSEC-CELEC+CSCAT
      RETURN
      END
      SUBROUTINE ION1(V,CION)
      COMMON /RSPEED/SPEED,ASPEED,BSPEED,CSPEED
      COMMON /GENRAL/EMAX,DELMAX,TSEC,RION,TION,RPI,CUTOFF,CUTLOW,KBRNCH
      W=V/TION
      CION=0.
      IF(W.GT.10.)GO TO 2
      IF(V.GE.0.)GO TO 1
      U=-W
      CION=RION*(ASPEED+BSPEED*U**CSPEED)
      GO TO 2
    1 CION=RION*EXP(-W)
    2 RETURN
      END
      SUBROUTINE ION2(V,CION)
      COMMON /GENRAL/EMAX,DELMAX,TSEC,RION,TION,RPI,CUTOFF,CUTLOW,KBRNCH
      W=V/TION
      CION=0.
      IF(W.GT.10.)GO TO 2
      IF(V.GE.0)GO TO 1
      U=SQRT(-W)
      CION=RION*(RPI*U+EXP(-W)*ERFC(U))
      GO TO 2
    1 CION=RION*EXP(-W)
    2 RETURN
      END
      SUBROUTINE ION3(V,CION)
      COMMON /RSPEED/SPEED,ASPEED,BSPEED,CSPEED
      COMMON /GENRAL/EMAX,DELMAX,TSEC,RION,TION,RPI,CUTOFF,CUTLOW,KBRNCH
      W=V/TION
       CION=0.
      IF(W.GT.10.)GO TO 2
      IF(V.GE.0.)GO TO 1
      U=-W
      CION=RION*ASPEED
      GO TO 2
    1 CION=RION*EXP(-W)
    2 RETURN
      END
      SUBROUTINE RETARD(V,CUR,TEMP)
      COMMON /GENRAL/EMAX,DELMAX,TSEC,RION,TION,RPI,CUTOFF,CUTLOW,KBRNCH
      BETA=-V/TEMP
      BETA1=-BETA
      IF(ABS(BETA).GT.100.)GO TO 1
      IF(KBRNCH.EQ.3)CUR=CUR*EXP(BETA)
      IF(KBRNCH.EQ.2)CUR=CUR*EXP(BETA)*(RPI*SQRT(BETA1)+EXP(BETA1)*
     1ERFC(SQRT(BETA1)))
      IF(KBRNCH.EQ.1)CUR=CUR*EXP(BETA)*(1.+BETA1)
      GO TO 2
    1 CUR=0.
    2 RETURN
      END
      SUBROUTINE SCAT1(V,CSCAT,M,H,Y)
      DIMENSION H(M),Y(M)
      REAL KON,KON1,KON2,KON3,KON4,KON5,KON6,KON7,KON10,KON20,KON30,
     1KON40,KON50,KON60,KON70,KON11,KON21,KON31,KON41,KON51,KON61,KON71,
     1KON12,KON22,KON32,KON42,KON52,KON62,KON72
     1,KON13,KON23,KON33,KON43,KON53,KON63,KON73,KON8,KON80,KON81,KON
     182,KON83
      COMMON /GENRAL/EMAX,DELMAX,TSEC,RION,TION,RPI,CUTOFF,CUTLOW,KBRNCH
      COMMON /SCAT/WEIGHT(5),SCATA(5),SCATB(5),SCATC(5),VSCAT(5),NUMPAR
      AI=22.30259-10./CUTOFF 50.*CUTLOW
      TBAR=(18.0819-26.6666/*CUTLOW**1.5-20./SQRT(CUTOFF))/
     1(85.23904-400.*SQRT(CUTLOW)/3-20./(3.*CUTOFF**1.5))
      CSCAT=0.
      DO 80 K=1,NUMPAR
      A=SCATA(K)
      B=SCATB(K)
      C=SCATC(K)
      VBAR=VSCAT(K)
      EBAR=VBAR*TBAR
      U=-V
      IF(U.LT.10.)GO TO 2
      FXSUM=0.
      KON1=(CUTOFF-U)/2.
      KON2=(CUTOFF+U)/(CUTOFF-U)
      DO 10 I=1,M
      KON3=Y(I)+1.
      KON5=KON1*KON3
      KON6=Y(I)+KON2
      FX=H(I)*KON3*(A-B*EXP(-C*KON5))/KON6**3
      FXSUM=FX+FXSUM
   10 CONTINUE
      FXSUM=10.*FXSUM/KON1
      GO TO 6
    2 IF(U.LT.1.)GO TO 3
      FXSUM=0.
      KON1=(CUTOFF-10.)/2.
      KON2=(CUTOFF+10.)/(CUTOFF-10.)
      KON4=KON2-2.*U/(CUTOFF-10.)
      KON7=V/KON1
      KON10=(10.-U)/2.
      KON20=(10.+U)/(10.-U)
      DO 30 I=1,M
      KON3=Y(I)+KON4
      KON5=KON3*KON1
      KON6=Y(I)+KON2
      KON30=Y(I)+1.
      KON50=KON10*KON30
      KON60=Y(I)+KON20
      FX=H(I)*(KON3*10.*(A-B*EXP(-C*KON5))/(KON1*KON6**3)+
     1KON30*(A-B*EXP(-C*KON50))/KON60**2)
      FXSUM=FX+FXSUM
   30 CONTINUE
      GO TO 6
    3 IF(U.LT..1)GO TO 4
      FXSUM=0.
      KON1=(CUTOFF-10.)/2.
      KON2=(CUTOFF+10.)/(CUTOFF-10.)
      KON4=KON2-2.*U/(CUTOFF-10.)
      KON7=V/KON1
      KON10=(11.-2.*U)/9.
      KON30=11./9.
      KON70=V/4.5
      KON11=(1.-U)/2.
      KON21=(1.+U)/(1.-U)
      DO 50 I=1,M
      KON3=Y(I)+KON4
      KON5=KON3*KON1
      KON6=Y(I)+KON2
      KON40=Y(I)+KON10
      KON20=4.5*(Y(I)+KON10)
      KON60=Y(I)+KON30
      KON31=Y(I)+1.
      KON51=KON31*KON11
      KON61=Y(I)+KON21
      FX=H(I)*(KON3*10.*(A-B*EXP(-C*KON5))/(KON1*KON6**3)+
     1KON20*(A-B*EXP(-C*KON20))/(4.5*KON60**2)
     1+(A-B*EXP(-C*KON51))/(KON11*KON61**3))
      FXSUM=FX+FXSUM
   50 CONTINUE
      GO TO 6
    4 IF(U.LT.CUTLOW)GO TO 5
      FXSUM=0.
      KON1=(CUTOFF-10.)/2.
      KON2=(CUTOFF+10.)/(CUTOFF-10.)
      KON4=KON2-2.*U/(CUTOFF-10.)
      KON7=V/KON1
      KON10=(11.-2.*U)/9.
      KON30=11./9.
      KON70=V/4.5
      KON11=(1.1-2.*U)/.9
      KON31=1.1/.9
      KON71=V/.45
      KON12=(.1-U)/2.
      KON22=(.1+U)/(.1-U)
      DO 70 I=1,M
      KON3=Y(I)+KON4
      KON5=KON3*KON1
      KON6=Y(I)+KON2
      KON40=Y(I)+KON10
      KON20=4.5*(Y(I)+KON10)
      KON60=Y(I)+KON30
      KON41=Y(I)+KON11
      KON61=Y(I)+KON31
      KON21=.45*(Y(I)+KON11)
      KON32=KON12*(Y(I)+1.)
      KON62=Y(I)+KON22
      FX=H(I)*(KON3*10.*(A-B*EXP(-C*KON5))/(KON1*KON6**3)+
     1KON20*(A-B*EXP(-C*KON20))/(4.5*KON60**2)+
     1(A-B*EXP(-C*KON21))*KON21/(.2025*KON61**3)+
     1100.*(A-B*EXP(-C*KON32))*KON32/KON62)
      FXSUM=FX+FXSUM
   70 CONTINUE
      GO TO 6
    5 IF(U.LT.0.)GO TO 8
      FXSUM=0.
      KON1=(CUTOFF-10.)/2.
      KON2=(CUTOFF+10.)/(CUTOFF-10.)
      KON4=KON2-2.*U/(CUTOFF-10.)
      KON7=V/KON1
      KON10=(11.-2.*U)/9.
      KON30=11./9.
      KON70=V/4.5
      KON11=(1.1-2.*U)/.9
      KON31=1.1/.9
      KON71=V/.45
      KON12=(.1-CUTLOW)/2.
      KON22=(.1+CUTLOW)/(.1-CUTLOW)
      KON32=KON22-U/KON12
      KON72=V/KON12
      KON13=(CUTLOW-U)/2.
      KON23=(CUTLOW+U)/(CUTLOW-U)
      DO 90 I=1,M
      KON3=Y(I)+KON4
      KON5=KON3*KON1
```

```
      KON6=Y(I)+KON2
      KON40=Y(I)+KON10
      KON20=4.5*(Y(I)+KON10)
      KON60=Y(I)+KON30
      KON41=Y(I)+KON11
      KON61=Y(I)+KON31
      KON21=.45*(Y(I)+KON11)
      KON42=(Y(I)+KON32)*KON12
      KON52=Y(I)+KON32
      KON62=Y(I)+KON22
      KON43=Y(I)+1.
      KON33=KON13*KON43
      KON63=Y(I)+KON23
      FX=H(I)*(KON3*10.*(A-B*EXP(-C*KON5))/(KON1*KON6**3)+
     1KON20*(A-B*EXP(-C*KON20))/(4.5*KON60**2)+
     1(A-B*EXP(-C*KON21))*KON21/(.2025*KON61**3)+
     1100.*KON42*(A-B*EXP(-C*KON42))/KON62+
     1100.*(A-B*EXP(-C*KON33))*KON33*KON13/CUTLOW)
      FXSUM=FX+FXSUM
   20 CONTINUE
      GO TO 6
    4 FXSUM=0.
      KON1=(CUTOFF-10.)/2.
      KON2=(CUTOFF+10.)/(CUTOFF-10.)
      KON4=KON2-2.*U/(CUTOFF-10.)
      KON7=V/KON1
      KON10=(11.-2.*U)/9.
      KON30=11./9.
      KON70=V/4.5
      KON11=(1.1-2.*U)/.9
      KON31=1.1/.9
      KON71=V/.45
      KON12=(.1-CUTLOW)/2.
      KON22=(.1+CUTLOW)/(.1-CUTLOW)
      KON32=KON22-U/KON12
      KON72=V/KON12
      KON13=CUTLOW/2.
      KON23=V/KON13
      DO 100 I=1,M
      KON3=Y(I)+KON4
      KON5=KON3*KON1
      KON6=Y(I)+KON2
      KON40=Y(I)+KON10
      KON20=4.5*(Y(I)+KON10)
      KON60=Y(I)+KON30
      KON41=Y(I)+KON11
      KON61=Y(I)+KON31
      KON21=.45*(Y(I)+KON11)
      KON42=(Y(I)+KON32)*KON12
      KON52=Y(I)+KON32
      KON62=Y(I)+KON22
      KON63=Y(I)+1.
      KON53=KON63+KON23
      KON33=KON13*KON63
      KON43=KON13*KON53
      IF(KBRNCH.EQ.1)
     1FX=H(I)*(KON3*10.*(A-B*EXP(-C*KON5))/(KON1*KON6**3)+
     1KON20*(A-B*EXP(-C*KON20))/(4.5*KON60**2)+
     1(A-B*EXP(-C*KON21))*KON21/(.2025*KON61**3)+
     1100.*KON42*(A-B*EXP(-C*KON42))/KON62+
     1100.*KON43*KON13*(A-B*EXP(-C*KON43)))/CUTLOW)
      IF(KBRNCH.EQ.2)
     1FX=H(I)*(10.*(KON3*ARSIN(SORT(KON6/KON3))/KON6+SORT(KON7/KON6))
     1*(A-B*EXP(-C*KON5))/(KON6**2*KON1)+
     1(KON40*ARSIN(SORT(KON60/KON40))/KON60+SORT(KON70/KON60))
     1*(A-B*EXP(-C*KON20))/KON60+
     1(KON41*ARSIN(SORT(KON61/KON41))/KON61+SORT(KON71/KON61))
     1*(A-B*EXP(-C*KON21))/(.45*KON61**2)
     1+100.*(KON52*ARSIN(SORT(KON62/KON52))/KON62+SORT(KON72/KON62))*
     1(A-B*EXP(-C*KON42))*KON12+
     1100.*KON13*KON33*(A-B*EXP(-C*KON43))*(KON53*ARSIN(SORT(KON63/
     1KON53))/KON63+SORT(KON23/KON63)))/CUTLOW)
      IF(KBRNCH.EQ.3)
     1FX=H(I)*(10.*(A-B*EXP(-C*KON5))/(KON6**2*KON1)+
     1(A-B*EXP(-C*KON20))/KON60+
     1(A-B*EXP(-C*KON21))/(.45*KON61**2)+
     1100.*(A-B*EXP(-C*KON42))*KON12+
     1100.*KON13*KON33*(A-B*EXP(-C*KON43)))/CUTLOW)
      FXSUM=FX+FXSUM
  100 CONTINUE
      IF(KBRNCH.EQ.2)FXSUM=.6366198*FXSUM
      IF(U.LT.0.)CALL RETARD(V,FXSUM,EBAR)
    6 CSCAT=CSCAT+WEIGHT(K)*FXSUM
   80 CONTINUE
      CSCAT=CSCAT/A1
      RETURN
      END
      SUBROUTINE ELEC1(V,CELEC)
      COMMON /Z/H(100),Y(100),KI(5),M
      COMMON /GENRAL/EMAX,DELMAX,TSEC,RION,TION,RPI,CUTOFF,CUTLOW,KBRNCH
      REAL KON,KON1,KON2,KON3,KON4,KON5,KON6,KON7,KON10,KON20,KON30,
     1KON40,KON50,KON60,KON70,KON11,KON21,KON31,KON41,KON51,KON61,KON71,
     1KON12,KON22,KON32,KON42,KON52,KON62,KON72
     1,KON13,KON23,KON33,KON43,KON53,KON63,KON73,KON8,KON80,KON81,KON
     182,KON83
      A1=22.30259-10./CUTOFF-50.*CUTLOW
      U=-V
      IF(CUTLOW.LT.1E-6)KBRNCH=3
      IF(V.GT.0.)GO TO 5
      IF(U.GT.CUTOFF)CELEC=0.
      IF(U.LT.10.)GO TO 2
      CELEC=5./U+5.*U/CUTOFF**2-10./CUTOFF
      GO TO 6
    2 IF(U.LT.1.)GO TO 3
      CELEC=ALOG(10./U)+U/20.+5.*U/CUTOFF**2-10./CUTOFF
      GO TO 6
    3 IF(U.LT..1)GO TO 4
      CELEC=ALOG(10.)+.5/U-10./CUTOFF+5.*U/CUTOFF**2-.45*U
      GO TO 6
    4 IF(U.LE.CUTLOW)GO TO 55
      CELEC=20.-150.45*U+5.*U/CUTOFF**2-10./CUTOFF-100.*U*ALOG(.1/U)+
     1ALOG(10.)
      GO TO 6
   55 CELEC=20.-50.45*U+5.*U/CUTOFF**2-10./CUTOFF-100*CUTLOW-100.*U*ALOG
     1(.1/CUTLOW)+ALOG(10.)+50.*(CUTLOW-U)**2/CUTLOW
      GO TO 6
    5 IF(KBRNCH.EQ.3)CELEC=A1
      IF(KBRNCH.EQ.1)CELEC=20.-50.45*U+5.*U/CUTOFF**2-10./CUTOFF
     1-100.*CUTLOW-100.*U*ALOG(.1/CUTLOW)+ALOG(10.)+50.*CUTLOW-100.*U
      IF(KBRNCH.NE.2)GO TO 6
      FXSUM=0.
      KON1=12.732395/(CUTOFF-10.)
      KON2=(CUTOFF+10.)/(CUTOFF-10.)
      KON3=2.*V/(CUTOFF-10.)
      KON4=KON2+KON3
      KON10=.6366198
      KON30=V/4.5
      KON40=1.222222+KON30
      KON11=1.414711
      KON31=V/.45
      KON41=1.222222+KON31
      KON12=31.830988/(.1-CUTLOW)
      KON22=(.1+CUTLOW)/(.1-CUTLOW)
      KON32=2.*V/(.1-CUTLOW)
      KON42=KON22+KON32
      KON13=15.91549*CUTLOW
      KON33=2.*V/CUTLOW
      KON43=1.+KON33
      DO 50 I=1,M
      KON5=Y(I)+KON4
      KON6=Y(I)+KON2
      KON50=Y(I)+KON40
      KON60=Y(I)+1.222222
      KON51=Y(I)+KON41
      KON61=Y(I)+1.222222
      KON52=Y(I)+KON42
      KON62=Y(I)+KON22
      KON53=Y(I)+KON43
      KON63=Y(I)+1.
      FX=H(I)*(KON1*(KON5*ARSIN(SORT(KON6/KON5))/KON6+SORT(KON3/KON6))
     1/KON6**2+KON10*(KON50*ARSIN(SORT(KON60/KON50))/KON60+SORT(KON30
     1/KON60))/KON60+KON11*(KON51*ARSIN(SORT(KON61/KON51))/KON61+
     3SORT(KON31/KON61))/KON61**2+KON12*(KON52*ARSIN(SORT(KON62/KON52))/
     4KON62+SORT(KON32/KON62))+KON13*(KON53*ARSIN(SORT(KON63/KON53))+
     5+SORT(KON33*KON63)))
      FXSUM=FX+FXSUM
   50 CONTINUE
      CELEC=FXSUM
    6 CELEC=CELEC/A1
      RETURN
      END
      SUBROUTINE SEC1(V,CSEC,M,H,Y)
      DIMENSION H(M),Y(M)
      COMMON /GENRAL/EMAX,DELMAX,TSEC,RION,TION,RPI,CUTOFF,CUTLOW,KBRNCH
      REAL KON,KON1,KON2,KON3,KON4,KON5,KON6,KON7,KON10,KON20,KON30,
     1KON40,KON50,KON60,KON70,KON11,KON21,KON31,KON41,KON51,KON61,KON71,
     1KON12,KON22,KON32,KON42,KON52,KON62,KON72
     1,KON13,KON23,KON33,KON43,KON53,KON63,KON73,KON8,KON80,KON81,KON
     182,KON83
      A1=22.30259-10./CUTOFF-50.*CUTLOW
      U=-V
      KON=7.4*DELMAX/EMAX
      IF(U.LT.0.)GO TO 5
      IF(U.LT.10.)GO TO 1
      FXSUM=0.
      KON1=10.*KON
      KON2=(CUTOFF+U)/(CUTOFF-U)
      KON3=2.*(CUTOFF-U)/EMAX
      DO 10 I=1,M
      KON5=Y(I)+1.
      KON7=SORT(KON3*KON5)
      KON6=Y(I)+KON2
      FX=H(I)*KON5**2*EXP(-KON7)/KON6**3
      FXSUM=FX+FXSUM
   10 CONTINUE
      CSEC=KON1*FXSUM
      GO TO 6
    1 IF(U.LT.1.)GO TO 2
      FXSUM=0.
      KON1=10.*KON
      KON2=(CUTOFF+10.)/(CUTOFF-10.)
      KON3=2.*(CUTOFF-10.)/EMAX
      KON4=(CUTOFF+10.-2.*U)/(CUTOFF-10.)
      KON10=(10.-U)*KON/2.
      KON20=(10.+U)/(10.-U)
      KON30=2.*(10.-U)/EMAX
      DO 20 I=1,M
      KON5=Y(I)+KON4
      KON6=Y(I)+KON2
      KON7=SORT(KON3*KON5)
      KON50=Y(I)+1.
      KON60=Y(I)+KON20
      KON70=SORT(KON30*KON50)
      FX=H(I)*(KON1*KON5**2*EXP(-KON7)/KON6**3+KON10*KON50**2*
     1EXP(-KON70)/KON60**2)
      FXSUM=FX+FXSUM
   20 CONTINUE
      CSEC=FXSUM
      GO TO 6
    2 IF(U.LT..1)GO TO 3
      FXSUM=0.
      KON1=10.*KON
      KON2=(CUTOFF+10.)/(CUTOFF-10.)
      KON3=2.*(CUTOFF-10.)/EMAX
      KON4=(CUTOFF+10.-2.*U)/(CUTOFF-10.)
      KON10=4.5*KON
      KON30=18./EMAX
      KON40=1.222222-U/4.5
      KON11=KON
      KON21=(1.+U)/(1.-U)
      KON31=2.*(1.-U)/EMAX
      DO 30 I=1,M
      KON5=Y(I)+KON4
      KON6=Y(I)+KON2
      KON7=SORT(KON3*KON5)
      KON50=Y(I)+KON40
      KON60=Y(I)+1.222222
      KON70=SORT(KON30*KON50)
      KON51=Y(I)+1.
      KON61=Y(I)+KON21
      KON71=SORT(KON31*KON51)
      FX=H(I)*(KON1*KON5**2*EXP(-KON7)/KON6**3+KON10*KON50**2*
     1EXP(-KON70)/KON60**2+KON11*KON51**2*EXP(-KON71)/KON61**3)
      FXSUM=FX+FXSUM
   30 CONTINUE
      CSEC=FXSUM
      GO TO 6
    3 IF(U.LT.CUTLOW)GO TO 4
      FXSUM=0.
      KON1=10.*KON
      KON2=(CUTOFF+10.)/(CUTOFF-10.)
      KON3=2.*(CUTOFF-10.)/EMAX
      KON4=(CUTOFF+10.-2.*U)/(CUTOFF-10.)
      KON10=4.5*KON
      KON30=18./EMAX
      KON40=1.222222-U/4.5
      KON11=KON
      KON31=1.8/EMAX
      KON41=1.222222-U/.45
      KON12=25.*KON*(.1-U)**2
      KON22=(.1+U)/(.1-U)
      KON32=2.*(.1-U)/EMAX
      DO 40 I=1,M
      KON5=Y(I)+KON4
      KON6=Y(I)+KON2
      KON7=SORT(KON3*KON5)
      KON50=Y(I)+KON40
      KON60=Y(I)+1.222222
      KON70=SORT(KON30*KON50)
      KON51=Y(I)+KON41
      KON61=Y(I)+1.222222
      KON71=SORT(KON31*KON51)
      KON52=Y(I)+1
      KON62=Y(I)+KON22
```

```
      KON72=SQRT(KON32*KON52)
      FX=H(I)*(KON1*KON5**2*EXP(-KON7)/KON6**3+KON10*KON50**2*
     1EXP(-KON70)/KON60**2+KON11*KON51**2*EXP(-KON71)/KON61**3
     1+KON12*KON52**2*EXP(-KON72)/KON62)
      FXSUM=FX+FXSUM
   40 CONTINUE
      CSEC=FXSUM
      GO TO 6
    4 FXSUM=0.
      KON1=10.*KON
      KON2=(CUTOFF+10.)/(CUTOFF-10.)
      KON3=2.*(CUTOFF-10.)/EMAX
      KON4=(CUTOFF+10.-2.*U)/(CUTOFF-10.)
      KON10=4.5*KON
      KON30=18./EMAX
      KON40=1.222222-U/4.5
      KON11=KON
      KON31=1.8/EMAX
      KON41=1.222222-U/.45
      KON12=25.*KON*(.1-CUTLOW)**2
      KON22=(.1+CUTLOW)/(.1-CUTLOW)
      KON32=2.*(.1-CUTLOW)/EMAX
      KON42=KON22-2.*U/(.1-CUTLOW)
      KON8=2.*V/(CUTOFF-10.)
      KON80=V/4.5
      KON81=V/.45
      KON82=2.*V/(.1-CUTLOW)
      KON13=12.5*KON*(CUTLOW-U)**3/CUTLOW
      KON23=(CUTLOW+U)/(CUTLOW-U)
      KON33=2.*(CUTLOW-U)/EMAX
      DO 50 I=1,M
      KON5=Y(I)+KON4
      KON6=Y(I)+KON2
      KON7=SQRT(KON3*KON5)
      KON50=Y(I)+KON40
      KON60=Y(I)+1.222222
      KON70=SQRT(KON30*KON50)
      KON51=Y(I)+KON41
      KON61=Y(I)+1.222222
      KON71=SQRT(KON31*KON51)
      KON52=Y(I)+KON42
      KON62=Y(I)+KON22
      KON72=SQRT(KON32*KON52)
      KON53=Y(I)+1.
      KON63=Y(I)+KON23
      KON73=SQRT(KON33*KON53)
      FX=H(I)*(KON1*KON5**2*EXP(-KON7)/KON6**3
     1+KON10*KON50**2*EXP(-KON70)/KON60**2+KON11*KON51**2*EXP(-KON71)
     1/KON61**3+KON12*KON52**2*EXP(-KON72)/KON62)
     1+KON13*KON53**2*EXP(-KON73))
      FXSUM=FX+FXSUM
   50 CONTINUE
      CSEC=FXSUM
      GO TO 6
    5 KON1=10.*KON
      KON2=(CUTOFF+10.)/(CUTOFF-10.)
      KON3=2.*(CUTOFF-10.)/EMAX
      KON4=(CUTOFF+10.-2.*U)/(CUTOFF-10.)
      KON10=4.5*KON
      KON30=18./EMAX
      KON40=1.222222-U/4.5
      KON11=KON
      KON31=1.8/EMAX
      KON41=1.222222-U/.45
      KON12=25.*KON*(.1-CUTLOW)**2
      KON22=(.1+CUTLOW)/(.1-CUTLOW)
      KON32=2.*(.1-CUTLOW)/EMAX
      KON42=KON22-2.*U/(.1-CUTLOW)
      KON8=2.*V/(CUTOFF-10.)
      KON80=V/4.5
      KON81=V/.45
      KON82=2.*V/(.1-CUTLOW)
      KON83=2.*V/CUTLOW
      KON13=12.5*KON*CUTLOW**2
      KON33=2.*CUTLOW/EMAX
      KON43=1.-2.*U/CUTLOW
      FXSUM=0.
      DO 60 I=1,M
      KON5=Y(I)+KON4
      KON6=Y(I)+KON2
      KON7=SQRT(KON3*KON5)
      KON50=Y(I)+KON40
      KON60=Y(I)+1.222222
      KON70=SQRT(KON30*KON50)
      KON51=Y(I)+KON41
      KON61=Y(I)+1.222222
      KON71=SQRT(KON31*KON51)
      KON52=Y(I)+KON42
      KON62=Y(I)+KON22
      KON72=SQRT(KON32*KON52)
      KON53=Y(I)+KON43
      KON73=SQRT(KON33*KON53)
      KON63=Y(I)+1.
      IF(KBRNCH.EQ.1)FX=H(I)*(KON1*KON5**2*EXP(-KON7)/KON6**3
     1+KON10*KON50**2*EXP(-KON70)/KON60**2+KON11*KON51**2*EXP(-KON71)
     1/KON61**3+KON12*KON52**2*EXP(-KON72)/KON62)
     1+KON13*KON53**2*EXP(-KON73))
      IF(KBRNCH.EQ.2)FX=H(I)*(KON1*KON5*(KON5*ARSIN(SQRT(KON6/KON5))/KON
     16+SQRT(KON8/KON6))*EXP(-KON7)/KON6**2+KON10*KON50*(KON50*ARSIN(SQR
     2T(KON60/KON50))/KON60+SQRT(KON80/KON60))*EXP(-KON70)/KON60+KON11
     2*KON51*(KON51*ARSIN(SQRT(KON61/KON51))/KON61+SQRT(KON81/KON61))
     3*EXP(-KON71)/KON61**2+KON12*KON52*(KON52*ARSIN(SQRT(KON62/KON52)
     4)/KON62+SQRT(KON82/KON62))*EXP(-KON72)
     5+KON13*KON53*KON63*(KON53*ARSIN(SQRT(KON63/KON53))/KON63+
     6SQRT(KON83/KON63))*EXP(-KON73))
      IF(KBRNCH.EQ.3)FX=H(I)*(KON1*KON5*EXP(-KON7)/KON6**2
     1+KON10*KON50*EXP(-KON70)/KON60+KON11*KON51*EXP(-KON71)/KON61**2
     1+KON12*KON52*EXP(-KON72))+KON13*KON53*KON63*EXP(-KON73)
      FXSUM=FX+FXSUM
   60 CONTINUE
      IF(KBRNCH.EQ.2)FXSUM=.6366198*FXSUM
      CSEC=FXSUM
      IF(U.LT.0.)CALL RETARD(V,CSEC,TSEC)
    6 CSEC=CSEC/A1
    8 RETURN
      END
$ENTRY
1414 4
-.9862838          .3511946E-01
-.9284349          .8015809E-01
-.8272013          .1215186
-.6872929          .1572032
-.5152486          .1855384
-.3191124          .2051985
- .1080549         .2152639
  .1080549         .2152639
 .3191124          .2051985
 .5152486          .1855384
 .6872929          .1572032
 .8272013          .1215186
 .9284349          .8015809E-01
 .9862838          .3511946E-01
```

```
   .075        1.         .003
2111
  0.          50.        .001
  0.            1.         1.          1.
1.         .3        1
1.         .10       0.         0.          .464
```

Appendix B: Listing of threshold temperatures program

```
F 00004 IS ON CR00131 USING 00024 BLKS R=0000

FN4
      PROGRAM TEMP

  THIS S/P COMPUTES THE CRITICAL TEMPERATURES FOR SPACECRAFT
  MATERIALS INCLUDING ANGULAR DEPENDENCE OF YIELD.

      DOUBLE PRECISION DZ
      COMMON/MTRL/EMX(32),DMX(32),BSA(32),BSB(32),BSC(32),Z(32)
      COMMON EM,DM,REM,A,B,C,FZ,IFLG,ICHK,DINTG(10)
      DIMENSION TC(32,6),DUMMY(32),IDUM(64),DDTS(2)
      EQUIVALENCE (DUMMY,IDUM)
      DATA EPS/1.0E-5/,DDTS/0.05,1.0/
      DATA NUMS/1/,NUME/29/,IFLGE/3/,NP1/1/,NP2/2/
      DATA LU/13/,ITRAC/20/,ISECT/0/

      DO 500 N=NUMS,NUME
      WRITE(1,800) N
  800 FORMAT(5( * ),'N=',I2)
      EM=EMX(N)
      DM=DMX(N)
      DO 500 NPART=NP1,NP2
      FAC=NPART*2-3
      DTS=DDTS(NPART)
      WRITE(1,804) NPART
  804 FORMAT(10'.','NPART=',I2)
      DO 500 IFLG=1,IFLGE
      IF(IFLG.GE.2) GO TO 90
      A=0.0
      B=0.0
      C=0.0
      IF(NPART.EQ.1) TS=EM
      IF(NPART.EQ.2) TS=EM/2.0
      GO TO 10
   90 A=BSA(N)
      B=BSB(N)
      C=BSC(N)
      IF(IFLG.EQ.2) GO TO 6
      DZ=DBLE(Z(N))
      BETAB=SNGL(7.37D0*DZ**(-0.56875D0))
      FZ=(EXP(BETAB)-BETAB-1.0)/BETAB**2*2.0
      WRITE(1,201) Z(N),BETAB
  201 FORMAT(10X,'Z=',F5.1,5X,'BETAB=',E15.7)
      GO TO 8
    6 FZ=1.0
    8 TS=TC(N,NPART)
      IF(NPART.EQ.1.AND.TS.LE.1.0E-4) TS=EM
      IF(NPART.EQ.2.AND.TS.LE.1.0E-4) TS=EM/2.0
   10 DT=DTS
      NPATH=1
   20 FL=RCUR(TS)
   30 TSS=TS+DT*FAC
      IF(TSS.LE.1.0E-4) GO TO 500
      FH=RCUR(TSS)
      WRITE(1,600) TS,FL,FH,DT
  600 FORMAT(4F15.7)
      IF(NPART.EQ.1.AND.TS.LE.1.0E-4) GO TO 500
      IF(NPART.EQ.2.AND.TS.GT.20.0) GO TO 500
      IF(FL*FH) 50,50,40
   40 TS=TS+DT*FAC
      FL=FH
      GO TO (30,50),NPATH
   50 DT=DT/2.0
      NPATH=2
      IF(DT-EPS) 60,60,30
   60 NPAT=NPART+(IFLG-1)*2
      TC(N,NPAT)=TS
      WRITE(1,200) IFLG,TS
  200 FORMAT(5X,'IFLG=',I1,5X,'TC=',F8.4)
  500 CONTINUE
      CALL EXEC(2,LU,EMX,64,ITRAC,ISECT)
      CALL EXEC(2,LU,DMX,64,ITRAC,ISECT+1)
      CALL EXEC(2,LU,BSA,64,ITRAC,ISECT+2)
      CALL EXEC(2,LU,BSB,64,ITRAC,ISECT+3)
      CALL EXEC(2,LU,BSC,64,ITRAC,ISECT+4)
      CALL EXEC(2,LU,Z,64,ITRAC,ISECT+5)
      DO 300 J=1,6
      ISEC=ISECT+J+5
      DO 310 I=1,32
      DUMMY(I)=TC(I,J)
  310 CONTINUE
      CALL EXEC(2,LU,DUMMY,64,ITRAC,ISEC)
  300 CONTINUE
      IDUM(1)=NUMS
      IDUM(2)=NUME
      CALL EXEC(2,LU,IDUM,64,ITRAC,ISECT+12)
      STOP
      END

      BLOCK DATA
      COMMON/MTRL/EMX(32),DMX(32),BSA(32),BSB(32),BSC(32),Z(32)
      DATA EMX/0.8,0.3,0.3,0.42,0.33,0.35,0.3,0.4,0.3,0.15,
     A0.25,0.8,0.25,0.4,0.8,0.35,0.36,0.35,0.39,0.85,0.15,
     B0.41,0.15,0.15,0.15,0.72,0.48,0.59,0.3/
      DATA DMX/1.45,0.97,2.6,2.5,3.46,0.75,2.2,5.0,3.0,2.1,
     A1.8,1.4,0.92,4.0,1.0,2.55,2.39,2.35,1.52,6.38,2.1,
     B2.05,2.1,2.1,2.1,1.03,1.49,1.86,1.0/
      DATA Z/79.0,13.0,10.0,10.0,10.0,6.0,29.0,29.0,8.0,5.3,
     A5.3,24.4,12.0,10.0,47.0,15.3,16.2,16.9,15.3,10.0,5.0,
     B10.0,5.0,5.0,5.0,70.1,42.0,63.4,6.0/
      DATA BSA/0.4802,0.1568,0.1238,0.1238,0.1238,0.08,0.3136,
     A0.3136,0.09,0.07,0.07,0.275,0.146,0.1238,0.39,0.183,0.192,
     B0.2,0.183,0.1238,0.06,0.1238,0.06,0.06,0.06,0.456,0.373,
     C0.438,0.08/
      DATA BSB/0.3566,0.0303,0.0172,0.0172,0.0172,0.0,0.0692,
     A0.0692,0.0,0.0,0.0,0.06,0.025,0.0172,0.289,0.037,0.04,
     B0.042,0.037,0.0172,0.0,0.0172,0.0,0.0,0.0,0.338,0.276,
     C0.325,0.0/
      DATA BSC/0.6103,0.3431,0.3435,0.3435,0.3435,0.0,0.6207,
     A0.6207,0.0,0.0,0.0,0.54,0.344,0.3435,0.632,0.382,0.399,
     B0.41,0.382,0.3435,0.0,0.3435,0.0,0.0,0.0,0.612,0.617,
     C0.613,0.0/
      END

      SUBROUTINE INTG(EINTG)
      COMMON EM,DM,REM,A,B,C,FZ,IFLG,ICHK,DINTG(10)
      DATA ERROR/1.0E-4/,EFCUT/1.0E-7/
      ITER=0
      DO 100 I=1,200
      CUTOF=I
      FCUT=FINTG(CUTOF)
      IF(FCUT.LE.EFCUT) GO TO 110
  100 CONTINUE
      WRITE(1,989)
  989 FORMAT('INSUFFICIENT # OF DOMAINS')
  110 DE=CUTOF/10.0
      SUM1=FCUT
      SUMA=0.0
      SUM2=0.0
      E=DE
    5 SUM2=SUM2+FINTG(E)
      E=E+DE
      IF(E+DE-2.0*CUTOF) 5,50,4
    4 SUM4=0.0
      E=DE/2.0
   10 SUM4=SUM4+FINTG(E)
      E=E+DE
      IF(E-CUTOF) 10,50,20
C
   20 EINTG=DE/6.0*(SUM1+2.0*SUM2+4.0*SUM4)
      ITER=ITER+1
      II=ITER+1
      IF(II.GE.6) GO TO 60
      DINTG(II)=EINTG
      GO TO 40
   60 DINTG(4)=DINTG(5)
      DINTG(5)=EINTG
C
   40 IF(ABS(EINTG-SUMA)/EINTG-ERROR) 50,50,30
C
   30 SUM2=SUM2+SUM4
      SUMA=EINTG
      DE=DE/2.0
      GO TO 4
C
   50 DINTG(1)=CUTOF
      DINTG(2)=FCUT
      DINTG(3)=DE
      IF(ITER.EQ.1) RETURN
      EINTG=(16.0*DINTG(5)-DINTG(4))/15.0
      RETURN
      END
C
      FUNCTION RCUR(T)
      COMMON EM,DM,REM,A,B,C,FZ,IFLG,ICHK,DINTG(10)
      REM=EM/T
      CALL INTG(EINTG)
      RCUR=REM**2*EXP(2.0)*DM*EINTG*(A-B/(1.0+C*T)**2)*FZ-1.0
      RETURN
      END
C
      FUNCTION FINTG(E)
      COMMON EM,DM,REM,A,B,C,FZ,IFLG,ICHK,DINTG(10)
      P1=-2.0*SQRT(E)-REM*E
      IF(IFLG.GE.3) GO TO 10
      IF(P1.GT.-20.0) GO TO 5
      FINTG=0.0
      RETURN
    5 FINTG=E**2*EXP(P1)
      RETURN
   10 X=0.2755*(ALOG(E)-1.658)
      Y=-0.0228/(X+SQRT(X**2+0.0228))
      BETAS=EXP(Y)
      P2=P1*BETAS
      IF(P1.GT.-20.0) GO TO 15
      FINTG=(E/BETAS)**2*EXP(P2)*2.0
      RETURN
   15 FINTG=(E/BETAS)**2*(EXP(P2)-EXP(P1)*(BETAS+1.0))*2.0
      RETURN
      END
      END$
```

C1

Appendix C: Listing of CYLVIA

```
FTN4
(HMS BLK1,0)
      PROGRAM CYVIA(3,150)

C  PURPOSE OF PROGRAM CYLVIA
C
C        TO OBTAIN THE POTENTIAL DISTRIBUTION AROUND A SPACECRAFT
C        UNDER MAGNETOSPHERIC CONDITIONS
C        THE SPACECRAFT IS DIVIDED INTO SECTORS, EACH OF WHICH
C        MAY CONTAIN SEVERAL GRID POINTS. SURFACE MATERIAL PROPERTIES
C        ARE SET UP IN S/P DATRED USING THE CONTENT OF THE FILE DATBAZ.
C       THE INPUT FILE DATIN, WHICH SAYS HOW THE SPACE CRAFT SURFACE IS
C       DIVIDED,IS ALSO READ IN DATRED.
C  FILE DESCRIPTIONS:
C   ---------- -----
C        DATIN,DATBAZ,POTIN ARE INPUT FILES
C        POTIN IS USED IF FLAG IPOIS=2
C        POTIN MAY CONTAIN SURFACE POTENTIAL INFORMATION
C        WHICH SHORT-CIRCUITS THE  GUESS FIELD  INITIALIZATION.
C        TAPE8=POTOUT CONTAINS BRIEF INFORMATION ABOUT POTENTIAL
C        TAPE8 IS REWOUND AT EACH MAJOR ITERATION
C        TAPE9=OUTPUT IS THE MAIN OUTPUT FILE
C
C
C  DESCRIPTION OF THE VARIABLES
C  ----------- -- --- ---------
C        ACURCY TOLERANCE FOR OBJECTIVE FUNCTION
C        ANG    ANGULAR LOCATION OF GRID POINT(USED IN S/P PHOTO)
C        ANGO IS ANGLE BETWEEN THE FIRST GRID POINT AND THE SUBSOLAR POIN
C        CBSCAT  (NON-DIM) CURRENT DUE TO BACKSCATTERS
C        CBSCA1  EMITTED BACKSCATTER CURRENT AT GRID POINT
C        BSCAT1,BSCAT2,BSCAT3  COEFFICIENTS OF BACKSCATTER EMISSION
C        CELEC ELECTRON CURRENT AT GRID POINT
C        CGUN(144)   CURRENT DUE TO ELECTRON OR ION GUN
C        CION ION CURRENT
C        CPHOMX  PHOTOELECTRON COEFF FOR A GIVEN MATERIAL(M/CBDAMF M##2)
C        CPHOTO PHOTOELECTRON CURRENT
C        CSEC1  EMITTED SECONDARY ELECTRON CURRENT AT GRID POINTS
C        CSEC TOTAL SECONDARY ELECTRON CURRENT AT GRID POINTS
C        CSECTR TOTAL CURRENT FOR A GIVEN SECTOR
C        DELEF,DELMAX COEFFICIENTS OF SECONDARY EMISSION
C        DELT  STEP IN ORBIT INTEGRATION
C        DPHI POTENTIAL INCREMENT
C        DENE1 ELECTRON NUMBER DENSITY PER CUBIC METER
C        CORRESPONDING TO A LOW ENERGY MAXWELLIAN VELOCITY DISTRIBUTION
C        DENE2 ELECTRON NUMBER DENSITY CORRESPONDING TO A HIGH ENERGY
C        DENI1,DENI2 ION NUMBER DENSITIES
C        DELTAX  GRID SPACING IN X COORDINATE
C        DELTAY  GRID SPACING IN Y COORDINATE
C        DVDX=POTENTIAL DERIVATIVE IN X DIRECTION
C        DVDY=POTENTIAL DERIVATIVE IN Y DIRECTION
C        EV= 8TH ORDER MONOENERGETIC VELOCITIES
C        EK=8TH ORDER MONOENERGETIC WEIGHTS
C        GAMMA  TEMPERATURE RATIO(GIVEN SPECIES TEMP/REF TEMP)
C        IDTL  FLAG FOR FINE STRUCTURE (SECTORS)
C       IFAULT=FLAG FOR INCLUDING OR EXCLUDING IONS,ELECTRONS,PHOTONS,ETC
C       IGEOM IS FLAG FOR GEOMETRY OF SPACECRAFT
C       IGEOM=1 THE CRAFT IS AN INFINTE CYLINDER
C       IGEOM=2 THE CRAFT IS A PROLATE SPHEROID
C        ILEV FACTOR OF DIVISION OF VELOCITY LEVELS
C        IPOIS IS A FLAG USED IN S/P GENGRD
C        IPOIS=0  A GUESS FIELD IS COMPUTED USING LOCAL CURRENT BALANCE
C        IPOIS=1  POISSON EQUATION SOLVED USING UPDATED DATA
C        IPOIS=2  RESTART THE PROGRAM AT THE LAST ITERATION
C        ISTABLE IS FLAG INDICATING STABILITY FROM ONE ITERATION TO NEXT
C        ITER IS NUMBER OF MAJOR ITERATIONS
C        ISYM=0  CORRESPONDS TO NO AXIAL SYMMETRY
C        ISYM=2  THE FIRST GRID POINT MAKES AN ANGLE EQUAL TO ANGO WITH
C                THE SUBSOLAR POINT
C        JLOCAL DEFINES SURFACE MATERIAL
C        IROOT=FLAG INDICATING CONVERGENCE AT A GIVEN GRID POINT
C        H=INITIAL STEP IN ORBIT INTEGRATION
C        LUN3 LOGICAL UNIT 3(DATBAZ)
C        LUN4 LOGICAL UNIT 4(DETAILS FOR CURRENT DUE TO SECONDARIES)
C        LUN5 LOGICAL UNIT 5(DATIN)
C        LUN6 LOGICAL UNIT 6(CUT-OFF BOUNDARIES  -ACCEPTANCE ANGLES
C        FOR PRIMARY PARTICLES
C        LUN7 LOGICAL UNIT 7(MISCELLENOUS)
C        LUN8 LOGICAL UNIT 8(POTOUT)
C        LUN9 LOGICAL UNIT 9(OUTPUT)
C        LUN10 LOGICAL UNIT 10(POTIN)
C        M=NUMBER OF UNKNOWNS IN I DIRECTION (RADIAL)
C        NPOINT=NUMBER OF UNKNOWNS IN J DIRECTION (ANGLE)
C        NANG THE VELOCITY SPACE IS DIVIDED INTO 2#NANG-1 INTERVALS
C        FOR ANGLES(USED FOR SECONDARY ARRIVING PARTICLES)
C        NLEVEL NUMBER OF VELOCITY LEVELS, =NTIME#NSAMP
C        NTIMES FACTOR OF MULTIPLICATION OF NUMBER OF VELOCITY LEVELS
C        NSAMP=NUMBER OF VELOCITY LEVELS AT OUR DISPOSAL.
C        NSECT=NUMBER OF SECTORS
C        PERSEC=NUMBER OF GRID POINTS PER SECTOR
C        PSECTR=SURFACE POTENTIAL FOR A GIVEN SECTOR
C        PBOUND=POTENTIAL ON OUTER BOUNDARY (USUALLY ZERO)
C        PHI=POTENTIAL OF SPACECRAFT SURFACE AT GRID POINTS
C        RATIO=Q(1)T(2)/Q(2)T(1). SEE SUBROUTINE GENRAT
C        RAD=RADIUS OF SPACECRAFT
C        SBOUND=LN(OUTER BOUNDARY RADIUS/SPACECRAFT RADIUS)
C       SECINI CONTAINS INDICES OF GRID POINTS PER SECTOR
C       SECPR1=EMITTED SECONDARY CURRENT DUE TO PROTONS
C       SECPRN=TOTAL SECONDARY CURRENT DUE TO PROTONS
C        SUNANG IS ANGLE OF SUN RELATIVE TO HORIZONTAL AXIS
C        TEMPB=BSCATTER ELECTRON TEMPERATURE IN EV
C        TEMPE1=PRIMARY ELECTRON TEMPERATURE IN EV
C        TEMPE2=PRIMARY ELECTRON TEMPERATURE IN EV CORRESPONDING
C               TO HIGHER ENERGY POPULATION
C        TEMPI1=ION TEMPERATURE IN EV
C        TEMPI2=ION TEMPERATURE IN EV FOR HIGH ENERGY SPECIES
C        TEMPP= PHOTELECTRON TEMP IN EV
C        TEMPR=REFERENCE TEMPERATURE TAKEN AS 1000EV
C        TEMPS=SECONDARY ELECTRON TEMP IN EV
C        TEMPSP=TEMPERATURE OF SECONDARY ELECTRONS PRODUCED BY
C               PROTON IMPACTS
C        XS,XSS  RADIAL DISTANCE
C        Y=SURFACE ANGLE(DEFINED IN GENORD)
C IMPORTANT CONSTANTS DEFINED ELSEWHERE
C---------------------------------------
C ACCMIN IS TOLERANCE FOR ACCEPTANCE ANGLE(DEFINED IN S/P GENORD)
C HCONS DEFINES INITIAL STEP SIZE FOR ORBITS (S/P GENGRD AND PROLAT)
C TOL IS TOLERANCE FOR ORBIT INTEGRATION (DEFINED IN S/P ORBIT)
C VACC IS ACCURACY FOR VELOCITY SPACE MINIMA (S/P ION AND ELEC)
C EXTERNALS
C--------------
C        BSCATR
C        DATE
C        DATRED
C        DIVPOT
C        ELEC
C        ELPRN
C        FINORD
C        FLWOP
C        GENGRD
C        GENRAT
C        GENVEL
C        GUN
C        ION
C        INSECT
C        INTEOR
C        PHOTO
C        PROLAT
C        SECON
C  I/O
C ---
C        LOGICAL UNIT 8
C        LOGICAL UNIT 9
C***************************************************************
C
C---
C  NOTE: IN GENERAL, INDEX I REFERS TO SECTORS
C                    INDEX J REFERS TO GRID POINTS (ANGLES)
C                    INDEX K REFERS TO VELOCITY LEVELS
C                    INDEX L REFERS TO ANGLES (VELOCITY SPACE)
C
C
       IMPLICIT DOUBLE PRECISION (A-H,O-Z)
       INTEGER PERSEC,SECIND,PERPAN,PANIND
       COMMON/BLK1
      1    PSAVE(16,10),CSAVE(16,10),PSECTR(16),CSECTR(16),
      1DELEE(20),DELMAX(16),EMAX(16),ETA(20),BSCAT1(16),
      2BSCAT2(16),BSCAT3(16),PHO(20,40),
      1ANG(40),CPHOMX(16),SUNANG,ANGO,BOUNDL(20,40),BOUNDI(20,40),
      2ANGL(20,40),VII(20,40),ICHECK(20,40),RATIO(13),IROOT(16),
      3IFAULT(14),JLOCAL(16),ICON(16),
      1VI(20),EV(8),NLEV,NTIMES,ILEV,TIMIN(20),TIMAX(20),VINUM(20),
      1IFLG,NPOINT,M,IGEOM,ISYM,IPOIS,NSECT,NPEROD,MPEROD,IDIMY,
      2GAMMA,DVDX(70,16),DVDY(70,16),Y(16),DELTAX,DELTAY,
      3YS(66,16),XS(66),XSS(70),
      1VARBLE(4),T,DELT,SBOUND,NDIM,POT,H,J1,RPI,RPI2,RPIHAF,SQTPI,
      2SAY,TEMPE1,TEMPE2,DENE1,DENE2,DENI1,DENI2,TEMPI1,TEMPI2,
      3TEMPR,LUN3,LUN4,LUN5,LUN6,LUN7,LUN8,LUN9,LUN10,
      1IDTL(16),JDTL(16),PERSEC(16),SECIND(16),IPANEL(16),
      2PERPAN(16,4),PANIND(16),
      1CION(16),CELEC(16),CSEC(16),CSEC1(16),CPHOTO(16),
      2CPHOT1(16),CBSCAT(16),CBSCA1(16),SECPRN(16),SECPR1(16),
      3PHI(16),CGUN(16),
      1NASE,JOML,DPHI,NANG,RAD,PBOUND,OMEGA,ALPHA,SO,JP,IS,ECS,
      2EAPVI2(20),EFTVI(20),IERROR,TETMIN,TETMAX,BETA1,BETA2,JLESS,
      3L2,CARRIV,CPH,XI,CSECEL,CSECE1,CS,CB,CS1,LB1
C
       COMMON/BLK2/ DELP(16),EXMP(16)
       COMMON/BLK3/INAM1,INAM2,INAM3,INAM4,INAM5,INAM6,INAM7,INAM8,
      1 INAM9,INAM10,INAM11,INAM12,INAM13,INAM14,INAM15,INAM16,
      2 INAM17,INAM18,INAM19,INAM20
       COMMON/BLK4/CE,CURENT
C*****************************************************************
C
       COMMON/BLK6/TEMPE,TEMPI,DEN,RATE,NUM,NTOUR,VZERO,K,TZERO,VMIN,
      1 VMED
       DIMENSION NAM6(3),NAM7(3)
       DIMENSION NAME1(3),NAME2(3),NAME3(3),NAME4(3),NAMES(3)
       DIMENSION NAM13(3),NAM14(3),NAM15(3),NAM16(3)
       DATA NAME1/2HDA,2HTR,2HD /
       DATA NAME2/2HPO,2HIS,2HS /
       DATA NAME3/2HIO,2HNC,2HC /
       DATA NAME4/2HPH,2HOT,2HO /
       DATA NAME5/2HGU,2HES,2HS /
       DATA NAM6/2HDI,2HVP,2HT /
       DATA NAM7/2HSU,2HMC,2HR /
       DATA NAM13/2HEL,2HEC,2HC /
       DATA NAM14/2HSE,2HCO,2HN /
       DATA NAM15/2HBS,2HCA,2HT /
       DATA NAM16/2HEL..,PR,2HN /
       LUN3=13
       LUN4=1
       LUN5=5
       LUN6=1
       LUN7=1
       LUN8=1
       LUN9=1
       LUN10=10
C BRANCHING INSTRUCTIONS FOR SEGMENTATION
       IF(INAM1.EQ.1) GO TO 9001
       IF(INAM2.EQ.1) GO TO 9002
       IF(INAM3.EQ.1) GO TO 9003
       IF(INAM4.EQ.1) GO TO 9004
       IF(INAM5.EQ.1) GO TO 9005
       IF(INAM6.EQ.1) GO TO 9006
       IF(INAM7.EQ.1) GO TO 9007
       IF(INAM13.EQ.1) GO TO 9013
       IF(INAM14.EQ.1) GO TO 9014
       IF(INAM15.EQ.1) GO TO 9015
       IF(INAM16.EQ.1) GO TO 9016
       LUN10=12
        RPI=3.14159260D0
        RPI2=2.0D0#RPI
        RPIHAF=RPI/2.0D0
        SQTPI=DSQRT(RPI)
        SAY=1.0D0/SQTPI
       ACUCY=1.0

        RAD=1.0
        PBOUND=0.0
        IFLG=0
        S0=0.0
        ILEV=1
       ITRACE=1

C*****************************************************************
C
C
C
C
C READ IN DATA AND ORGANIZE ESCTORS
        CALL EXEC(8,NAME1)
 9001 INAM1=0
        IDIMY=M+2
C
C
        N=NPOINT
C INITIALIZE ARRAYS
        DO 1  J=1,NPOINT
        IROOT(J)=0
        CION(J)=0.
        CELEC(J)=0.
        CPHOTO(J)=0.
        CSEC(J)=0.
        CBSCAT(J)=0.
        SECPRN(J)=0.
        CPHOT1(J)=0.
        CGUN(J)=0.
        CSECTR(J)=0.
        PHI(J)=0.0
      1 CONTINUE
C
```

```
C
C
C GENERATE INITIAL POTENTIAL SURFACE
      CALL EXEC(8,NAME1)
9001  INAM1=0
C
C
 1101 FORMAT (10X, NUMBER OF ITERATIONS= I3)
C
C COMPUTE NEW INDICES FOR CURRENT COLLECTION IF SYMMETRY
      NN=N
      IF(ISYM.EQ.2) NN=N/2
      NN1=NN-1
C
C
C BUILD POTENTIAL OF SECTORS FROM POTENTIAL AT GRID POINTS
C
      L=0
      DO 4000 I=1,NSECT
      K=PERSEC(I)
      N2=L+K
      N1=L+1
      DO 4001 NJ=N1,N2
      J=SECIND(NJ)
      PSECTR(I)=PHI(J)
 4001 CONTINUE
      L=L+K
 4000 CONTINUE
C
C****************************************************************************
C START ITERATION PROCESS
C
      ITER=0
 4010 NTOUR=0
C
C
C
 4199 CONTINUE
C GENERATE GRID AND CALL POISON SOLVER
      CALL EXEC(8,NAME2)
 9002 INAM2=0
 7221 FORMAT (8(1X,1PE12.5))
      CALL EXEC(8,NAM6)
 7006 INAM6=0
      NTOUR=NTOUR+1
      ITER=ITER+1
 4100 FORMAT(/' SECTOR    PSECTR     CSECTR       ELEC CUR')
C
C COMPUTE CURRENT COLLECTION FOR ALL SECTORS AND OBJECTIVE FUNCTION
C
C
C COMPUTE CURRENT COLLECTION DUE TO PRIMARY PARTICLES
C AND SECONDARY EMERGING PARTICLES:
C BYPASS COMPUTATIONS IF FLAGS IDTL AND JDTL CONCERNING FINE DETAIL
C STRUCTURE ARE NOT SET TO ONE
      L=0
      IS=0
 4200 IS=IS+1
      IF(IDTL(IS).NE.1) GO TO 4200
      KK=PERSEC(IS)
      N2=L+KK
      NJ=L
 4201 NJ=NJ+1
      J=SECIND(NJ)
      JP=J
      IF(JDTL(J).NE.1) GO TO 4201
      IF(ISYM.EQ.0) GO TO 4210
      IF(J.LE.NN) GO TO 4210
C IF THERE IS A SYMMETRY,SAVE COMPUTATIONS
C
      CION(J)=CION(N-J+1)
      CELEC(J)=CELEC(N-J+1)
      CSEC1(J)=CSEC1(N-J+1)
      CBSCA1(J)=CBSCA1(N-J+1)
      SECPR1(J)=SECPR1(N-J+1)
      GO TO 4202
C
 4210 CONTINUE
       CALL EXEC(8,NAME3)
 9003 INAM3=0
      CALL EXEC(8,NAM13)
 9013 INAM13=0
 4202 IF(NJ.LT.N2) GO TO 4201
      L=L+KK
      IF(IS.LT.NSECT) GO TO 4200
C
C
C COMPUTE CURRENT COLLECTION DUE TO SECONDARY ARRIVING PARTICLES
C
      L=0
      IS=0
 4250 IS=IS+1
      IF(IDTL(IS).NE.1) GO TO 4250
      KK=PERSEC(IS)
      N2=L+KK
      NJ=L
 4251 NJ=NJ+1
      J=SECIND(NJ)
      JP=J
      IF(JDTL(J).NE.1) GO TO 4251
      IF(ISYM.EQ.0) GO TO 4260
      IF(J.LE.NN) GO TO 4260
C IF THERE IS SYMMETRY,SAVE COMPUTATIONS
      CPHOTO(J)=CPHOTO(N-J+1)
      CSEC(J)=CSEC(N-J+1)
      CBSCAT(J)=CBSCAT(N-J+1)
      SECPRN(J)=SECPRN(N-J+1)
      GO TO 4252
C
 4260 CONTINUE
C COMPUTE SECONDARY CURRENTS
      IF(IFAULT(5).EQ.1) CALL EXEC(8,NAME4)
 9004 INAM4=0
      IF(IFAULT(7).EQ.1) CALL EXEC(8,NAM14)
 9014 INAM14=0
      IF(IFAULT(8).EQ.1) CALL EXEC(8,NAM15)
 9015 INAM15=0
      IF(IFAULT(14).EQ.1) CALL EXEC(8,NAM16)
 9016 INAM16=0
 4252 IF(NJ.LT.N2) GO TO 4251
      L=L+KK
      IF(IS.LT.NSECT) GO TO 4250
C
C
C COMPUTE TOTAL CURRENT AND SAVE TOTAL ELECTRON CURRENT FOR SECTOR
      IS=0
      WRITE(LUN4,4101)
 4101 FORMAT(//' J   POTENTIAL    TOTCUR       IONCUR    ELECCUR ',
     1'PHOTCUR   EMIPHOTCUR    SECONDCUR   EMISECUR   BSCATCUR'/
     2  EMIBSCAT   ELE IND  EMI ELEC IN     //)
      WRITE(LUN9,1101) ITER
 4270 IS=IS+1
```

```
      CALL EXEC(8,NAM[illegible])
      [illegible]CONTINUE
      FORMAT [illegible]1PE12.5
      WRITE(LUN9,4[illegible]) [illegible]PSECTR(I)[illegible]CSECTR(I)[illegible]
      IF [illegible]NSECT[illegible] GO TO 4[illegible]
      WRITE (LUN9,4100)
C
C BUILD OBJECTIVE FUNCTION AND SAVE CURRENTS AND POTENTIALS
C
      OBJ=0.0
      DO 7000 I=1,NSECT
      IF(IDTL(I).NE.1) GO TO 7000
      OBJ=OBJ+CSECTR(I)**2
      CSAVE(I,NTOUR)=CSECTR(I)
      PSAVE(I,NTOUR)=PSECTR(I)
 7000 CONTINUE
C
      WRITE(LUN9,7010) ITER,OBJ
 7010 FORMAT(1X,' ITERATION # ',I3,'   THE OBJECTIVE FUNCTION IS ',1PE12.5)
C
C
C
C IF ACCURACY OF OBJECTIVE FUNCTION IS OBTAINED,GO TO NEXT STEP
      IF(OBJ.LE.ACUCY) GO TO 4700
C
C
      IF(NTOUR.EQ.1) GO TO 7400
C CHECK IF CURRENTS HAVE CHANGED SIGN
C IROOT=1)
      DO 7210 I=1,NSECT
      IF(IROOT(I).EQ.1) GO TO 7210
      IF(CSAVE(I,NTOUR)*CSAVE(I,NTOUR-1).LE.0.0) IROOT(I)=1
 7210 CONTINUE
C
C CHECK IF CURRENTS OF ALL SECTORS HAVE CHANGED SIGN
      K=0
      DO 7220 I=1,NSECT
      IF(IROOT(I).EQ.1) K=K+1
 7220 CONTINUE
C
      IF(K.EQ.NSECT) WRITE(9,7230)
 7230 FORMAT(' ALL CURRENTS HAVE CHANGED SIGN, NOW REFINE SEARCH VECTORS
     1 ')
      IF(K.NE.NSECT) GO TO 7400
C
C FIND APPROXIMATE ROOTS FOR THE CURRENTS
C
      DO 7250 I=1,NSECT
      IF(IDTL(I).NE.1) GO TO 7250
      DO 7260 JJ=2,NTOUR
      JSTART=NTOUR-JJ+2
      IF(CSAVE(I,JSTART)*CSAVE(I,JSTART-1).LE.0.0) GO TO 7280
 7260 CONTINUE
C
      GO TO 7250
C
 7280 PSECTR(I)=(PSAVE(I,JSTART)-PSAVE(I,JSTART-1))/(CSAVE(I,JSTART)-
     1 CSAVE(I,JSTART-1))*(-CSAVE(I,JSTART-1))+PSAVE(I,JSTART-1)
C
 7285 CONTINUE
      WRITE(LUN9,7290) PSECTR(I),PSAVE(I,JSTART),PSAVE(I,JSTART-1),I
 7290 FORMAT(' APPROXIMATE ROOTS ARE',3(1X,1PD10.3),I6)
 7250 CONTINUE
C
C NOW DIVIDE INITIAL INCREMENT BY 5
      DPHI=DPHI/5.0
      NTOUR=0
C RESET FLAGS TO ZERO
      DO 7300 I=1,NSECT
      IROOT(I)=0
 7300 CONTINUE
      GO TO 4499
C
C MODIFY POTENTIAL OF SECTORS
 7400 CONTINUE
C
      DO 7450 I=1,NSECT
      IF(IDTL(I).NE.1) GO TO 7450
      IF(CSAVE(I,NTOUR).LE.0.0) PSECTR(I)=PSECTR(I)-DPHI
      IF(CSAVE(I,NTOUR).GT.0.0) PSECTR(I)=PSECTR(I)+DPHI
 7450 CONTINUE
C
C
C
C
C
C
C RESET POTENTIAL AT GRID POINTS
C
 4499 L=0
      DO 4500 I=1,NSECT
      IF(IDTL(I).EQ.0) GO TO 4500
      K=PERSEC(I)
      N2=L+K
      N1=L+1
      DO 4501 NJ=N1,N2
      J=SECIND(NJ)
      PHI(J)=PSECTR(I)
 4501 CONTINUE
      L=L+K
 4500 CONTINUE
C
C
 4540 CONTINUE
C
C
C
C
C REWIND SHORT FILE POTOUT
C CHECK SIZE OF ARRAYS
      IF(NTOUR.EQ.10) NTOUR=1
C
C
C RECOMPUTE CURRENTS
C
      GO TO 4199
C
C
 4700 JOML= JOML-1
      IF(JOML.EQ.0) WRITE (LUN9,4710)
 4710 FORMAT (2X,'COARSE CONVERGENCE HAS BEEN REACHED,DO ORBIT FOLLOWING
     1 FOR ALL PARTICLES')
      DO 4701 I=1,NSECT
      IROOT(I)=0
 4701 CONTINUE
      IF(JOML.EQ.0) GO TO 4010
      WRITE (LUN9,1301)
      WRITE (LUN8,1301)
 1301 FORMAT (1X, FINAL CONVERGENCE HAS BEEN REACHED )
C
C
      STOP
```

```
      END
      BLOCK DATA
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/BLK2/DELP(16),EXMP(16)
      COMMON/BLK3/INAM1,INAM2,INAM3,INAM4,INAM5,INAM6,INAM7,INAM8,
     1 INAM9,INAM10,INAM11,INAM12,INAM13,INAM14,INAM15,INAM16,
     2 INAM17,INAM18,INAM19,INAM20
      COMMON/BLK4/CE,CURENT
      COMMON/BLK5/TEMPE,TEMPI,DEN,RATE,NUM,NTOUR,VZERO,F,IZERO,VMIN,
     1 VMED
      DATA CE,CURENT/0.0D0,0.0D0/
      DATA DELP,EXMP/16*0.0D0,16*0.0D0/
      DATA INAM1,INAM2,INAM3,INAM4,INAM5/0,0,0,0,0/
      DATA INAM6,INAM7,INAM8,INAM9/0,0,0,0/
      DATA INAM10,INAM11,INAM12,INAM13,INAM14/0,0,0,0,0/
      DATA INAM15,INAM16,INAM17,INAM18,INAM19,INAM20/0,0,0,0,0,0/
      DATA TEMPE,TEMPI,DEN,RATE,VZERO,IZERO,VMIN,VMED/8*0.0D0/
      DATA NUM,NTOUR,K/3*0/
      END
      SUBROUTINE SATRED(KASE,NANG,JOML,DPHI)
C
C
C
C
C
C
C
C                     PACKAGE # 1
C                     ---------------
C
C READ IND DATA AND SURFACE PROPERTIES.ASSEMBLE SURFACE OF SATELLITE.
C
C
C
C
C
C
C
C
      INTEGER PERSEC,SECIND,PERPAN,PANIND
      COMMON PSAVE(144,10),CSAVE(144,10),PSECTR(144),CSECTR(144),
     1DELEE(20),DELMAX(144),EMAX(144),ETA(20),BSCAT1(144),
     2BSCAT2(144),BSCAT3(144),PHO(20,40),
     1ANG(40),CPHOMX(144),SUNANG,ANGO,BOUNDL(20,40),BOUNDI(20,40),
     2ANGL(20,40),VII(20,40),ICHECK(20,40),RATIO(13),IROOT(144),
     3IFAULT(14),JLOCAL(144),ICON(144),
     1VI(20),EV(8),NLEV,NTIMES,ILEV,TIMIN(20),TIMAX(20),VINUM(20),
     2A(144),B(144),C(144),BB(144),BW(144),TW(144),DW(144),
     3EW(144),TWOCOS(700),
     1IFLG,NPOINT,M,IGEOM,ISYM,IPOIS,NSECT,NPEROD,MPEROD,IDIMY,
     2GAMMA,DVDX(70,48),DVDY(70,48),Y(144),DELTAX,DELTAY,
     3YS(66,48),XS(66),XSS(70),
     1VARBLE(4),T,DELT,SBOUND,NDIM,POT,H,J1,RPI,RPI2,RPIHAF,SQTPI,
     2SAY,TEMPE1,TEMPE2,DENE1,DENE2,DENI1,DENI2,TEMPI1,TEMPI2,
     3TEMPR,LUN3,LUN4,LUN5,LUN6,LUN7,LUN8,LUN9,LUN10,
     1IDTL(144),JDTL(144),PERSEC(144),SECIND(144),IPANEL(144),
     2PERPAN(144,4),PANIND(144),
     1CION(144),CELEC(144),CSEC(144),CSEC1(144),CPHOTO(144),
     2CPHOT1(144),CBSCAT(144),CBSCA1(144),SECPRN(144),SECPRI(144),
     3PHI(144),CGUN(144),
     1KASE,JOML,DPHI,NANG,RAD,PBOUND,OMEGA,ALPHA,SO,JP,IS,ECS,
     2EXPVI2(20),EFTVI(20),IERROR,TETMIN,TETMAX,BETA1,BETA2,JLESS,
     3L2,CARRIV,CPH,XI,CSECEL,CSECE1,CS,CB,CS1,CB1
C
C**************************************************************************
C PURPOSE
C IN THIS S/P ,WE INPUT THE NECESSARY PARAMETERS FOR A CASE STUDY
C       KASE    :CASE STUDY
C       TEMPI1  :ION TEMPERATURE
C       TEMPI2  :ION TEMPERATURE
C       DENI1   :ION DENSITY
C       DENI2   :ION DENSITY
C       TEMPE1  :ELECTRON TEMPERATURE
C       TEMPE2  :ELECTRON TEMPERATURE
C       DENE1   :ECTRON DENSITY
C       DENE2   :ELECTRON DENSITY
C       SUNANG  :SUN DIRECTION WITH RESPECT TO ORIGIN OF  ANGLES
C       ANGO    :ANGLE BETWEEN FIRST GRID POINT AND ORIGIN OF ANGLES.
C       SBOUND:OUTER BOUNDARY
C       DPHI    :INCREMENT IN POTENTIAL IN KV
C       IPOIS   :FLAG FOR POISSON SOLVER
C       ISYM    :FLAG ABOUT SYMMETRY
C       IGEOM   :FLAG ABOUT GEOMETRY
C       NANG    : FOR ANGULAR SAMPLING FOR SECONDARY PARTICLES
C       NTIMES: FOR MULTIPLICATION OF VELOCITY LEVELS
C       NPOINT:NUMBER OF GRID POINTS IN ANGULAR DIRECTION
C       NSECT  :NUMBER OF SECTORS
C       SUNANG:ANGLE BETWEEN SUN DIRECTION AND ORIGIN
C       ANGO   :ANGLE BETWEEN FIRST GRID POINT AND ORIGIN
C       SBOUND:LOG OF RADIUS
C       DPHI   :INITIAL POTENTIAL INCREMENT(USED FOR CONVERGENCE)
C
C**************************************************************************
      CALL DATE (IDATE)
      READ(LUN5,99) KASE
   99 FORMAT (I3)
      WRITE (LUN9,990) KASE,IDATE
  990 FORMAT (40X,*KASE=*,I3,*      DATE=*,A10/40X,25(*-*)/////)
C
C
      READ(LUN5,5000) TEMPI1,TEMPI2,DENI1,DENI2,TEMPE1,TEMPE2,DENE1,
     1 DENE2
 5000 FORMAT(4E10.3)
      WRITE(LUN9,901) DENE1,DENE2,TEMPE1,TEMPE2,DENI1,DENI2,TEMPI1,TEMPI
     12
  901 FORMAT (1X,*  DENE1      DENE2        TEMPE1        TEMPE2     DENI1
     1    DENI2       TEMPI1       TEMPI2*/8(1X,1PE10.3)//)
C
C
      READ(LUN5,5001) SUNANG,ANGO,SBOUND,DPHI,IPOIS,ISYM,IGEOM,NANG,
     1 NTIMES,ILEV,JOML,NPOINT,M,NSECT
 5001 FORMAT(4E12.5,7I2,3I3)
      WRITE (LUN9,902) M,NPOINT,NSECT,IPOIS,NTIMES,NANG,ISYM,IGEOM,JOML
  902 FORMAT(1X,*    M NPOINT  NSECT  IPOIS  ITIMES NANG  ISYM IGEOM JOML
     1 */1X,9(I6)//)
      WRITE (LUN9,903) SUNANG,ANGO,SBOUND,DPHI
  903 FORMAT(1X,*  SUNANG      ANGO         SBOUND          DPHI*/4(1X,1PE10.3)//
     1)
C
C
C READ IN FLAGS FOR OPTIONS:
C-----------------------------
C       IFAULT(1)=1                               ION CURRENT IS COMPUTED
C       IFAULT(2)=1                               ELEC CUR IS COMPUTED
C       IFAULT(3)=1                               EMERGING SECONDARY CUR
C       IFAULT(4)=1                               EMERGING BSACT CUR IS C
C       IFAULT(5)=1,EMG PHOTO CUR COMPUTED
C       IFAULT(6)=1,RECEIVING PHOTO ELEC CUR
C       IFAULT(7)=1,RECV SEC CUR COMPUTED
C       IFAULT(8)=1,RECV BSACT CUR COMPUTED
C       IFAULT(9)=1,EMERG ELEC INDUCED BY PROTONS
C       IFAULT(10)=1,RECV ELEC INDUCED BY PROTONS
C       IFAULT(11).NE.ZERO, GUN IS WORKING
C       IFAULT(12)=1, FLOW EFFECT ON IONS
C       IFAULT(13)=1,FINE DETAIL IS SET UP
C       IFAULT(14)=1,LINEAR SPACE CHARGE IS SET UP
C
      READ(LUN5,101)(IFAULT(I),I=1,14)
  101 FORMAT(20I3)
      WRITE(LUN9,600)(IFAULT(J),J=1,14)
  600 FORMAT (1X,*FLAGS INDICATING WHETHER VARIOUS PARTICLES ARE INCLUDE
     1D (1) OR NOT INCLUDED (0)*/
     2 1X,*    ION ELEC SECD BACK PHOT  PHOT SECD BACK ELP  ELP  GUN FLOW
     1  DET  L.S.CHARGE* 1X,14I5//)
C
C READ IN SURFACE PROPERTIES AT GRID POINTS
C           -- -- ---- ----------- -----------------
C READ FIRST SURFACE MATERIAL
      WRITE (LUN9,601)
  601 FORMAT (1X,*TYPE OF SURFACE AT LOCATION J */
     11X*(1 IS GOLD,2 IS AL,3 IS AL OXIDE,4 IS QUARTZ,9 IS KAPTON)*)
C
      READ(LUN5,201) (JLOCAL(J),J=1,NPOINT)
  201 FORMAT(32I2)
      WRITE (LUN9,602) (JLOCAL(J),J=1,NPOINT)
  602 FORMAT(32I3)
C
C READ IN SURFACE CONDUCTIVITY PER SECTOR
C ICON=0 (INSULATOR),ICON=1 (CONDUCTOR)
      READ(LUN5,201) (ICON(I),I=1,NSECT)
      WRITE(LUN9,610)
  610 FORMAT(/1X,*FLAGS ABOUT CONDUCTIVITY OF SECTORS*)
      WRITE(LUN9,602) (ICON(I),I=1,NSECT)
C
C
C SET UP SURFACE PROPERTIES
C
C
      DO 100 J=1,NPOINT
      REWIND3
  110 READ(LUN3,301)JFIRST
  301 FORMAT(I2)
      IF(JLOCAL(J).NE.JFIRST)GO TO 110
      READ(LUN3,401)DELMAX(J),EMAX(J),CPHOMX(J)
  401 FORMAT(3X,3F10.3)
      READ(LUN3,501) BSCAT1(J),BSCAT2(J),BSCAT3(J)
  501 FORMAT(3X,4F10.3)
      READ(LUN3,502) DELP(J),EXMP(J)
  502 FORMAT(3X,2F10.3)
  100 CONTINUE
C
C
      RETURN
      END
      SUBROUTINE INSECT
C**************************************************************************
C INPUTS: NPOINT,NSECT
C OUTPUTS PERSEC,SECIND - VECTORS
      INTEGER I,J,K,L,TEMP(144)
C TEMP(...) MUST BE DIMENSIONED TO MAX NUMBER OF POINTS IN A SECTOR
C DATA FOR EACH SECTOR BEGINS ON A NEW LINE. FIRST NUMBER IS NUMBER
C OF POINTS IN SECTOR FOLLOWED BY INDICES OF THESE POINTS
C DATA FORMAT IS 16I5
C THIS S/P IS CONCERNED BY THE GEOMETRICAL ORGANIZATION OF THE
C DIFFERENT SECTORS. A CONDUCTING SECTOR MAY CONTAIN SEVERAL PANELS
C,WITH A MAXIMUM OF 4 PANELS PER SECTOR. A PANEL IS A SERIES OF
C  ADJACENT CONDUCTING GRID POINTS. WE HAVE ONE PANEL PER SECTOR
C FOR INSULATORS.
C IPANEL IS NUMBER OF PANELS PER SCETORS.
C PERPAN IS NUMBER OF GRID POINTS PER PANEL
C PANIND IS GRID POINTS INDICES STORED CLOCKWISE PER PANEL. THIS
C MAKES THE NUMERICAL INTEGRATION OF THE CURRENT EASIER.
C VALUES ARE STORED AS FOLLOWS: GRID POINTS OF THE FIRST PANEL
C (FIRST SECTOR)                                   EX: 31,32,1,2,......IT
C  ,THEN GRID POINTS OF THE SECOND PANEL(FIRST SECTOR)  ETC.
C**************************************************************************
C
      INTEGER PERSEC,SECIND,PERPAN,PANIND
      COMMON PSAVE(144,10),CSAVE(144,10),PSECTR(144),CSECTR(144),
     1DELEE(20),DELMAX(144),EMAX(144),ETA(20),BSCAT1(144),
     2BSCAT2(144),BSCAT3(144),PHO(20,40),
     1ANG(40),CPHOMX(144),SUNANG,ANGO,BOUNDL(20,40),BOUNDI(20,40),
     2ANGL(20,40),VII(20,40),ICHECK(20,40),RATIO(13),IROOT(144),
     3IFAULT(14),JLOCAL(144),ICON(144),
     1VI(20),EV(8),NLEV,NTIMES,ILEV,TIMIN(20),TIMAX(20),VINUM(20),
     2A(144),B(144),C(144),BB(144),BW(144),TW(144),DW(144),
     3EW(144),TWOCOS(700),
     1IFLG,NPOINT,M,IGEOM,ISYM,IPOIS,NSECT,NPEROD,MPEROD,IDIMY,
     2GAMMA,DVDX(70,48),DVDY(70,48),Y(144),DELTAX,DELTAY,
     3YS(66,48),XS(66),XSS(70),
     1VARBLE(4),T,DELT,SBOUND,NDIM,POT,H,J1,RPI,RPI2,RPIHAF,SQTPI,
     2SAY,TEMPE1,TEMPE2,DENE1,DENE2,DENI1,DENI2,TEMPI1,TEMPI2,
     3TEMPR,LUN3,LUN4,LUN5,LUN6,LUN7,LUN8,LUN9,LUN10,
     1IDTL(144),JDTL(144),PERSEC(144),SECIND(144),IPANEL(144),
     2PERPAN(144,4),PANIND(144),
     1CION(144),CELEC(144),CSEC(144),CSEC1(144),CPHOTO(144),
     2CPHOT1(144),CBSCAT(144),CBSCA1(144),SECPRN(144),SECPRI(144),
     3PHI(144),COUN(144),
     1KASE,JOML,DPHI,NANG,RAD,PBOUND,OMEGA,ALPHA,SO,JP,IS,ECS,
     2EXPVI2(20),EFTVI(20),IERROR,TETMIN,TETMAX,BETA1,BETA2,JLESS,
     3L2,CARRIV,CPH,XI,CSECEL,CSECE1,CS,CB,CS1,CB1
C
C**************************************************************************
C
      L=0
        DO 100 I=1,NSECT
        READ(LUN5,20)(K,(TEMP(J),J=1,K))
      WRITE(LUN9,40)I,(K,(TEMP(J),J=1,K))
   40 FORMAT(1X,*SECTOR NUMBER *,I2,* J VALUES ARE *,32I3)
   20   FORMAT((16I4))
        PERSEC(I)=K
          DO 90 J=1,K
          SECIND(J+L)=TEMP(J)
   90     CONTINUE
        L=L+K
  100   CONTINUE
      IF(L.NE.NPOINT)GO TO 99
C
C
        DO 60 I=1,NPOINT
        IF(SECIND(I).LE.0)GOTO 97
        IF(SECIND(I).GT.NPOINT)GOTO 96
   60   CONTINUE
      K=NPOINT-1
        DO 80 I=1,K
        L=I+1
          DO 70 J=L,NPOINT
          IF(SECIND(I).EQ.SECIND(J))GOTO 98
   70     CONTINUE
   80   CONTINUE
C
C
C
```

```
C
      READ(LUN5,20) (IPANEL(I),I=1,NSECT)
      DO 200 I=1,NSECT
C
      IP=IPANEL(I)
      READ(LUN5,20) (PERPAN(I,J),J=1,IP)
      WRITE(LUN9,901) I,IPANEL(I),(PERPAN(I,J),J=1,IP)
  200 CONTINUE
  901 FORMAT (1X,*SECTOR NUMBER *,I2,* NUMBER OF PANELS *,I2,
     1 * NR OF GRID POINTS PER PANEL*,4I4)
      READ(LUN5,20)(PANIND(J),J=1,NPOINT)
      IF(IFAULT(13).EQ.0) RETURN
      READ (LUN5,20) (IDTL(I),I=1,NSECT)
      WRITE(LUN9,900) (IDTL(I),I=1,NSECT)
  900 FORMAT(/,10X,*SECTORS WHERE WE WANT A FINE STRUCTURE*/(16I4))
C
      READ(LUN5,20) (JDTL(J),J=1,NPOINT)
      WRITE(LUN9,902) (JDTL(J),J=1,NPOINT)
  902 FORMAT(//10X,*GRID POINTS WHERE WE WANT A FINE STRUCTURE*,(16I6))
      RETURN
   99 WRITE(LUN9,30)
   30 FORMAT(* ERROR IN SECTOR DATA*,/,/)
      WRITE(LUN9,35)L,NPOINT
   35 FORMAT(* NUMBER OF INDICES =*,I3,* IS NOT EQUAL
     1TO NUMBER OF POINTS =*,I3)
C
C
      STOP
   98 WRITE(LUN9,30)
      WRITE(LUN9,31)I,J,SECIND(J)
   31 FORMAT(* INDEX(*,I3,*) = INDEX(*,I3,*) =*,I3)
C
C
      STOP
   97 WRITE(LUN9,30)
      WRITE(LUN9,32)I,SECIND(I)
   32 FORMAT(* INDEX(*,I3,*) =*,I3,* IS LESS THAN UNITY*)
C
C
      STOP
   96 WRITE(LUN9,30)
      WRITE(LUN9,33)I,SECIND(I),NPOINT
   33 FORMAT(* INDEX(*,I3,*) =*,I3,* IS GREATER THAN
     1THE NUMBER OF POINTS =*,I3)
C
C
      STOP
      END
      SUBROUTINE FINGRD
C
C PURPOSE:
C----------
C TO RESET ALL ARRAYS FOR A FINE DETAIL STRUCTURE OF SURFACE
C IN ORDER TO RESPECT GEOMETRY,INTERVALLES ARE DIVIDED BY THREE.
C
      INTEGER PERSEC,SECIND,PERPAN,PANIND
      COMMON PSAVE(144,10),CSAVE(144,10),PSECTR(144),CSECTR(144),
     1DELEE(20),DELMAX(144),EMAX(144),ETA(20),BSCAT1(144),
     2BSCAT2(144),BSCAT3(144),PHO(20,40),
     1ANG(40),CPHOMX(144),SUNANG,ANGO,BOUNDL(20,40),BOUNDI(20,40),
     2ANGL(20,40),VII(20,40),ICHECK(20,40),RATIO(13),IROOT(144),
     3IFAULT(14),JLOCAL(144),ICON(144),
     1VI(20),EV(8),NLEV,NTIMES,ILEV,TIMIN(20),TIMAX(20),VINUM(20),
     2A(144),B(144),C(144),BB(144),BW(144),TW(144),DW(144),
     3EW(144),TWOCOS(700),
     1IFLG,NPOINT,M,IGEOM,ISYM,IPOIS,NSECT,NPEROD,MPEROD,IDIMY,
     2GAMMA,DVDX(70,48),DVDY(70,48),Y(144),DELTAX,DELTAY,
     3YS(66,48),XS(66),XSS(70),
     1VARBLE(4),T,DELT,SBOUND,NDIM,POT,H,J1,RPI,RPI2,RPIHAF,SQTPI,
     2SAY,TEMPE1,TEMPE2,DENE1,DENE2,DENI1,DENI2,TEMPI1,TEMPI2,
     3TEMPR,LUN3,LUN4,LUN5,LUN6,LUN7,LUN8,LUN9,LUN10,
     1IDTL(144),JDTL(144),PERSEC(144),SECIND(144),IPANEL(144),
     2PERPAN(144,4),PANIND(144),
     1CION(144),CELEC(144),CSEC(144),CSEC1(144),CPHOTO(144),
     2CPHOT1(144),CBSCAT(144),CBSCA1(144),SECPRN(144),SECPR1(144),
     3PHI(144),CGUN(144),
     1KASE,JOML,DPHI,NANG,RAD,PBOUND,OMEGA,ALPHA,SO,JP,IS,ECS,
     2EXPVI2(20),EFTVI(20),IERROR,TETMIN,TETMAX,BETA1,BETA2,JLESS,
     3L2,CARRIV,CPH,XI,CSECEL,CSECE1,CS,CB,CS1,CB1
      EQUIVALENCE (IROOT(1),IDUMMY(1))
C
C
C*********************************************************************
C
C
      IOLD=NSECT
      JOLD=NPOINT
      NPOINT=3*JOLD
      ANGO=ANGO/3.0
      SUNANG=SUNANG/3.0
C
C FIND NEW FLAGS OF CORRESPONDING GRID POINTS
CFIRST SAVE OLD VALUES
C
       DO 100 J=1,JOLD
      IF(JDTL(J).EQ.0) GO TO 102
      IDUMMY(J*3-2)=1
      IDUMMY(J*3-1)=1
      IDUMMY(J*3)  =1
      GO TO 100
  102 IDUMMY(J*3-2)=0
      IDUMMY(J*3-1)=1
      IDUMMY(J*3  )=0
  100 CONTINUE
C
C
C
C NOW RESET ARRAY
      DO 110 J=1,NPOINT
  110 JDTL(J)=IDUMMY(J)
C
C
C
C FIND NEW INDICES OF CORRESPONDING GRID POINTS
      L=0
      II=0
      DO 112 I=1,IOLD
      K=PERSEC(I)
      N2=L+K
      N1=L+1
      DO 113 NJ=N1,N2
      J=SECIND(NJ)
      IF(ICON(I).EQ.0.AND.ISYM.EQ.2) GO TO 1100
      IDUMMY(NJ*3-2)=J*3-2
      IDUMMY(NJ*3-1)=J*3-1
      IDUMMY(NJ*3)  =J*3
      II=II+3
      GO TO 113
C
 1100 II=II+6
      IDUMMY(II-5)=J*3-2
      IDUMMY(II-4)=SECIND(NJ+1)*3
      IDUMMY(II-3)=J*3-1
      IDUMMY(II-2)=SECIND(NJ+1)*3-1
      IDUMMY(II-1)=J*3
      IDUMMY(II)=SECIND(NJ+1)*3-2
      GO TO 1111
  113 CONTINUE
C
 1111 CONTINUE
      L=L+K
  112 CONTINUE
C
C
C NOW RESET ARRAY
      DO 114 J=1,NPOINT
      SECIND(J)=IDUMMY(J)
  114 CONTINUE
C
C
C
C SET UP NEW INDICES OF GRID POINTS PER PANEL(PANIND)
      L=0
      II=0
      DO 251 I=1,IOLD
      K=PERSEC(I)
      N2=L+K
      N1=L+1
      DO 250 NJ=N1,N2
      J=PANIND(NJ)
      IF (ICON(I).EQ.0.AND.ISYM.EQ.2) GO TO 2500
      IDUMMY(NJ*3-2)=J*3-2
      IDUMMY(NJ*3-1)=J*3-1
      IDUMMY(NJ*3  )=J*3
      II=II+3
      GO TO 250
 2500 II=II+6
      IDUMMY(II-5)=J*3-2
      IDUMMY(II-4)=PANIND(NJ+1)*3
      IDUMMY(II-3)=J*3-1
      IDUMMY(II-2)=PANIND(NJ+1)*3-1
      IDUMMY(II-1)=J*3
      IDUMMY(II  )=PANIND(NJ+1)*3-2
      GO TO 2501
  250 CONTINUE
 2501 CONTINUE
      L=L+K
  251 CONTINUE
C
CRESET ARRAY
      DO 260 J=1,NPOINT
  260 PANIND(J)=IDUMMY(J)
C
C
C
C FIND NEW NUMBER OF POINTS (PERSEC) PER NEW SECTOR
C IF AN OLD SECTOR IS INSULATOR,IT IS BROKEN INTO 3 NEW SECTORS
C
      L=0
      II=0
      DO 120 I=1,IOLD
      K=PERSEC(I)
      IF (ICON(I).EQ.1) II=II+1
      IF (ICON(I).EQ.1) IDUMMY(II)=3*K
      IF (ICON(I).EQ.1) GO TO 120
       II=II+3
      IDUMMY(II-2)=1
      IF(ISYM.NE.0) IDUMMY(II-2)=2
      IDUMMY(II-1)=K
      IDUMMY(II  )=1
      IF (ISYM.NE.0) IDUMMY(II)=2
  120 CONTINUE
C
      NSECT=II
C
C
C NOW REORGANIZE ARRAYS
      DO 130 I=1,NSECT
  130 PERSEC(I)=IDUMMY(I)
C
C
C
C SET NEW FLAGS IDTL FOR SECTORS
      II=0
      DO 140 I=1,IOLD
      IF (ICON(I).EQ.1) II=II+1
      IF(ICON(I).EQ.1)  IDUMMY(II)=1
      IF(ICON(I).EQ.0.AND.IDTL(I).EQ.0) GO TO 150
      IF (ICON(I).EQ.0.AND.IDTL(I).EQ.1) GO TO 160
      GO TO 140
  150 II=II+3
      IDUMMY(II-2)=0
      IDUMMY(II-1)=1
      IDUMMY(II)=0
      GO TO 140
  160 II=II+3
      IDUMMY(II-2)=1
      IDUMMY(II-1)=1
      IDUMMY(II)=1
  140 CONTINUE
C
C
C NOW REORGANIZE ARRAY
      DO 170 I=1,NSECT
  170 IDTL(I)=IDUMMY(I)
C
C
C SET UP NEW NUMBER OF GRID POINTS PER PANEL
      II=0
      DO 270 I=1,IOLD
      IP=IPANEL(I)
      IF(ICON(I).EQ.1) II=II+1
      IF(ICON(I).EQ.1) IDUMMY(II)=IP
      IF(ICON(I).EQ.1) GO TO 280
      II=II+3
      IDUMMY(II-2)=1
      IF(ISYM.NE.0) IDUMMY(II-2)=2
      IDUMMY(II-1)=IP
      IDUMMY(II  )=1
      IF(ISYM.NE.0) IDUMMY(II)=2
      ID =IDUMMY(II-2)
      DO 275 L=1,ID
      IDUM(II-2,L)=1
  275 IDUM(II  ,L)=1
C
  280 DO 290 L=1,IP
      IF (ICON(I).EQ.1) IDUM(II,L)=PERPAN(I,L)
      IF(ICON(I).EQ.0) IDUM(II-1,L)=PERPAN(I,L)
  290 CONTINUE
  270 CONTINUE
C
C
C NOW RESET ARRAYS
      DO 295 I=1,NSECT
```

```
      IPANEL(I)=IDUMMY(I)
      IP=IPANEL(I)
      DO 295 L=1,IP
      PERPAN(I,L)=IDUM(I,L)
  295 CONTINUE
C
C
C
C CONDUCTIVITY OF NEW SECTORS
      II=0
      DO 300 I=1,IOLD
      IF (ICON(I).EQ.1) II=II+1
      IF (ICON(I).EQ.1) IDUMMY(II)=1
      IF(ICON(I).EQ.1) GO TO 300
      II=II+3
      IDUMMY(II-2)=0
      IDUMMY(II-1)=0
      IDUMMY(II  )=0
  300 CONTINUE
C
C
C NOW REORGANIZE ARRAYS
      DO 310 I=1,NSECT
      ICON(I)=IDUMMY(I)
  310 CONTINUE
C
C
C
C
C
C
C NOW RESET SURFACE PROPERTIES
C-----------------------
C FIRST SAVE OLD VALUES
      DO 450 J=1,JOLD
      IDUMMY(J)=JLOCAL(J)
  450 CONTINUE
C
      DO 460 J=1,JOLD
      JLOCAL(J*3-2)=IDUMMY(J)
      JLOCAL(J*3-1)=IDUMMY(J)
      JLOCAL(J*3  )=IDUMMY(J)
  460 CONTINUE
C
C
C
CFIND NEW SURFACE PROPERTIES
C
      DO 470 J=1,NPOINT
      REWIND LUN3
  480 READ(LUN3,500) JFIRST
  500 FORMAT (I2)
      IF(JLOCAL(J).NE.JFIRST) GO TO 480
      READ(LUN3,301) DELMAX(J),EMAX(J),CPHOMX(J)
  301 FORMAT(3X,3F10.3)
      READ(LUN3,301) BSCAT1(J),BSCAT2(J),BSCAT3(J)
      READ (LUN3,301) DELP(J),EXMP(J)
  470 CONTINUE
C
      WRITE(LUN9,900)
  900 FORMAT (//10X,*THE FINAL DETAIL STRUCTURE IS NOW SET UP*/10X,40
     1 (*-*))
C
C WRITE NEW VALUES
C-------------------
      WRITE (LUN9,901) (JDTL(J),J=1,NPOINT)
      WRITE(LUN9,902)
      WRITE(LUN9,901)  (SECIND(J),J=1,NPOINT)
      WRITE(LUN9,902)
      WRITE(LUN9,901) (PERSEC(I),I=1,NSECT)
      WRITE(LUN9,902)
      WRITE(LUN9,901) (IDTL(I),I=1,NSECT)
      WRITE(LUN9,902)
      WRITE(LUN9,901) (PANIND(J),J=1,NPOINT)
      WRITE(LUN9,902)
      WRITE(LUN9,901) (ICON(I),I=1,NSECT)
      WRITE(LUN9,902)
  902 FORMAT(/)
  901 FORMAT(32I2)
      DO 903 I=1,NSECT
      IP=IPANEL(I)
      WRITE(LUN9,902)
      WRITE(LUN9,901) (PERPAN(I,L),L=1,IP)
  903 CONTINUE
C
C
      RETURN
      END
      SUBROUTINE INTPOL(DUMMY)
C*********************************************************************
C PURPOSE: TO INTERPOLATE VARIABLES FOR FINEGRID STRUCTURE
C*********************************************************************
C
C
      DIMENSION  DUMMY(1)
      DIMENSION XX(144),YY(144),D(144)
      INTEGER PERSEC,SECIND,PERPAN,PANIND
      COMMON PSAVE(144,10),CSAVE(144,10),PSECTR(144),CSECTR(144),
     1DELEE(20),DELMAX(144),EMAX(144),ETA(20),BSCAT1(144),
     2BSCAT2(144),BSCAT3(144),PHO(20,40),
     1ANG(40),CPHOMX(144),SUMANG,ANGO,BOUNDL(20,40),BOUNDI(20,40),
     2ANGL(20,40),VII(20,40),ICHECK(20,40),RATIO(13),IROOT(144),
     3IFAULT(14),JLOCAL(144),ICON(144),
     1VI(20),EV(8),NLEV,NTIMES,ILEV,TIMIN(20),TIMAX(20),VIMUM(20),
     2A(144),B(144),C(144),BB(144),BW(144),TW(144),DW(144),
     3EW(144),TWOCOS(700),
     1IFLG,NPOINT,M,IGEOM,ISYM,IPOIS,NSECT,NPEROD,MPEROD,IDIMY,
     2GAMMA,DVDX(70,48),DVDY(70,48),Y(144),DELTAX,DELTAY,
     3YS(66,48),XS(66),XSS(70),
     1VARBLE(4),T,DELT,SBOUND,NDIM,POT,H,J1,RPI,RPI2,RPIHAF,SQTPI,
     2SAY,TEMPE1,TEMPE2,DENE1,DENE2,DENI1,DENI2,TEMPI1,TEMPI2,
     3TEMPR,LUN3,LUN4,LUN5,LUN6,LUN7,LUN8,LUN9,LUN10,
     1IDTL(144),JDTL(144),PERSEC(144),SECIND(144),IPANEL(144),
     2PERPAN(144,4),PANIND(144),
     1CION(144),CELEC(144),CSEC(144),CSEC1(144),CPHOTO(144),
     2CPHOT1(144),CBSCAT(144),CBSCA1(144),SECPRN(144),SECPR1(144),
     3PHI(144),CGUN(144),
     1RASE,JOML,DPHI,NANG,RAD,PBOUND,OMEGA,ALPHA,S0,JP,IS,ECS,
     2EXPVI2(20),EFTVI(20),IERROR,TETMIN,TETMAX,BETA1,BETA2,JLESS,
     3L2,CARRIV,CPH,XI,CSECEL,CSECE1,CS,CB,CS1,CB1
      EQUIVALENCE (XX(1),A(1)),(YY(1),BB(1)),(D(1),BW(1))
C
C IF SECTORS ARE CONDUCTORS,INTERPLOATION IS DONE PER PANEL
C IF SECTORS ARE INSULATORS,INTERPOLATION IS DONE BETWEEN ADJACENTS SECT
C
C DESCRIPTION  OF VARIABLES
C       JSTART:INDICE WHERE TO FIND FIRST GRID POINT OF SECTOR
C       XX AND YY:TEMPORAY STORAGE FOR INTERPOLATION
C B,C,D COEFFICIENT FOR SPLINE
C
      JSTART=1
      I=0
```

```
C LOOP OVER SECTORS
 1000 I=I+1
      IF(I.GT.NSECT) RETURN
C DO DIFFERENTS SHEMES IF INSULATORS OR CONDUCTORS
      IF(ICON(I).EQ.0) GO TO 10
C
C NOW DO INTERPOLATION FOR CONDUCTORS,FIND NB OF PANELS
C-------------------------------------------------------
      IP=IPANEL(I)
C FIND NB OF GRID POINTS PER PANEL AND ACCUMULATE FUNCTION VALUES
C WHEN JDTL(J)=1
C
      DO 2 I1=1,IP
      JP=PERPAN(I,I1)
      JEND=JP+JSTART-1
C
C NOW ACCUMUMLATE DATA
      JJ=0
      DO 3 J1=JSTART,JEND
      JJ=JJ+1
      J=PANIND(J1)
      IF (JDTL(J).EQ.1) XX(JJ)=DELTAY*(J1-1)
      IF(JDTL(J).EQ.1) YY(JJ)=DUMMY(J1)
    3 CONTINUE
C
C NOW CALL INTERPOLATION ROUTINE
C IF JJ=JP,NO INTERPOLATION NEEDED FOR THAT PANEL
      IF (JJ.EQ.JP) GO TO 8
C
      CALL SPLINE (JJ,XX,YY,B,C,D)
C
      DO 7 J1=JSTART,JEND
      J=PANIND(J1)
      IF(JDTL(J).EQ.0) DUMMY(J)=SPIDER(JJ,DELTAY*(J1-1),XX,Y,B,C,D)
    7 CONTINUE
C
    8 JSTART=JEND+1
    2 CONTINUE
      GO TO 1000
C
C
C INTERPOLATION FOR INSULATORS
C--------------------------------
   10 CONTINUE
C DO INTERPOLATION OE EXTRAPOLATION BETWEEN KNOWN SECTORS,
C TAKING INTO ACCOUNT THE SURFACE PROPERTIES OF SECTORS
C
      IF(IDTL(I).EQ.1) JSTART=JSTART+PERSEC(I)
      IF(IDTL(I).EQ.1) GO TO 1000
C
C SEE IF WE CAN DO AN INTERPOLATION
      J1=JSTART
      I1=I
   11 I1=I1-1
      IF(I1.LE.0) GO TO 20
      J1=J1-PERSEC(I1)
      IF(IDTL(I1).EQ.0) GO TO 11
      I2=I
      J2=JSTART
   12 I2=I2+1
      IF(I2.GT.NSECT) GO TO 30
      J2=J2+PERSEC(I2-1)
      IF(IDTL(I2).EQ.0) GO TO 12
C
C FUNCTION IS NOW KNOWN AT SECTORS I1 AND I2
C NOW CHECK IF SURFACE PROPERTIES ARE IDENTICAL AT I1 ANDI2
      IF (ICON(I1).NE.0) GO TO 20
      IF (ICON(I2).NE.0) GO TO 30
      JJ=SECIND(I)
      JJ1=SECIND(I1)
      JJ2=SECIND(I2)
      IF (JLOCAL(JJ1).NE.JLOCAL(JJ)) GO TO 20
      IF (JLOCAL(JJ2).NE.JLOCAL(JJ)) GO TO 30

   40 DUMMY(JJ)=(DUMMY(JJ2)-DUMMY(JJ1))/(Y(JJ2)-Y(JJ1))*(Y(JJ)-Y(JJ1))
     1+DUMMY(JJ1)
      IF(ISYM.EQ.2) DUMMY(NPOINT-JJ+1)=DUMMY(JJ)
      JSTART=JSTART+PERSEC(I)
      GO TO 1000
C
C DO EXTRAPOLATION FROM RIGHT OF I
   20 I1=I
      J1=JSTART
   21 I1=I+1
      J1=J1+PERSEC(I1-1)
      IF (ICON(I1).EQ.0) GO TO 21
      I2=I1
      J2=J1
   22 I2=I2+1
      J2=J2+PERSEC(I2-1)
      IF(ICON(I2).EQ.0) GO TO 22
      GO TO 40
C
C DO EXTRAPOLATION FROM LEFT OF I
   30 I1=I
      J1=JSTART
   31 I1=I-1
      J1=J1-PERSEC(I1)
      IF (ICON(I1).EQ.0) GO TO 31
      I2=I1
      J2=J1
   32 I2=I2-1
      J2=J2-PERSEC(I2)
      IF (ICON(I2).EQ.0) GO TO 32
      GO TO 40
C
      END
      SUBROUTINE SPLINE(N,X,Y,B,C,D)
C REF: MALCOM AND MOLLER
      DIMENSION X(N),Y(N),B(N),C(N),D(N)
C
C THE COEFFICIENTS B(I),C(I),AND D(I), I=1,2,....,N ARE COMPUTED
C FOR A CUBIC INTERPOLATING SPLINE
C
C S(X)=Y(I)+B(I)*(X-X(I))+C(I)*(X-X(I))**2+D(I)*(X-X(I))**3
C
C FOR X(I).LE.X(I+1)
C
C INPUT:
C N=NUMBER OF DATA POINTS
C X=ABSCISSA
C Y=ORDINATES
C
C OUTPUT:
C B,C,D=ARRAYS OF SPLINE COEFFICIENTS
C
C
      NM1=N-1
      IF(N.LT.2) STOP
      IF(N.LT.3) GO TO 50
C
C SET UP TRIDIAGONAL SYSTEM
```

C6

```
C
C B=DIAGONAL,D=OFFDIAGONAL,C=RIGHT HAND SIDE
C
      D(1)=X(2)-X(1)
      C(2)=(Y(2)-Y(1))/D(1)
      DO 10 I=2,NM1
      D(I)=X(I+1)-X(I)
      B(I)=2.0*(D(I-1)+D(I))
      C(I+1)=(Y(I+1)-Y(I))/D(I)
      C(I)=C(I+1)-C(I)
   10 CONTINUE
C
C END CONDITIONS. THIRD DERICATIVES AT X(1) AND X(N)
C ARE OBTAINED FROM DIVIDED DIFFERENCES
C
      B(1)=-D(1)
      B(N)=-D(N-1)
      C(1)=0.
      C(N)=0.
      IF(N.EQ.3) GO TO 15
      C(1)=C(3)/(X(4)-X(2))-C(2)/(X(3)-X(1))
      C(N)=C(N-1)/(X(N)-X(N-2))-C(N-2)/(X(N-1)-X(N-3))
      C(1)=C(1)*D(1)**2/(X(4)-X(1))
      C(N)=-C(N)*D(N-1)**2/(X(N)-X(N-3))
C
C FORWARD ELIMINATION
C
   15 DO 20 I=2,N
      T=D(I-1)/B(I-1)
      B(I)=B(I)-T*D(I-1)
      C(I)=C(I)-T*C(I-1)
   20 CONTINUE
C
C BACK SUBSTITUTION
C
      C(N)=C(N)/B(N)
      DO 30 IB=1,NM1
      I=N-IB
      C(I)=(C(I)-D(I)*C(I+1))/B(I)
   30 CONTINUE
C
C COMPUTE POLYNOMIAL COEFFICIENTS
C
      B(N)=(Y(N)-Y(NM1))/D(NM1)+D(NM1)*(C(NM1)+2.0*C(N))
      DO 40 I=1,NM1
      B(I)=(Y(I+1)-Y(I))/D(I)-D(I)*(C(I+1)+2.0*C(I))
      D(I)=(C(I+1)-C(I))/D(I)
      C(I)=3.0*C(I)
   40 CONTINUE
C
      C(N)=3.0*C(N)
      D(N)=D(N-1)
      RETURN
C
   50 B(1)=(Y(2)-Y(1))/(X(2)-X(1))
      C(1)=0.
      D(1)=0.
      B(2)=B(1)
      C(2)=0.
      D(2)=0.
      RETURN
      END
      FUNCTION SPIDER (N,U,X,Y,B,C,D)
      DIMENSION X(N),Y(N),B(N),C(N),D(N)
C
C THIS SUBROUTINE EVALUATE THE CUBIC SPLINE FUNCTION AT POINT U,USING
C HORNER ' S RULE
C IF U IS NOT IN THE PROPER INTERVAL, A BINARY SEARCH IS PERFORMED.
   10 I=1
      J=N+1
   20 K=(I+J)/2
      IF(U.LT.X(K)) J=K
      IF(U.GE.X(K)) I=K
      IF(J.GT.I+1) GO TO 20
C
C EVALUATE SPLINE
C
   30 DX=U-X(I)
      SPIDER =Y(I)+DX*(B(I)+DX*C(I)+DX*(D(I)))
      RETURN
      END
      SUBROUTINE GENRAT
C
C
C
C
C
C
C
C                   PACKAGE # 2
C                   -------------------
C
C                 GENERATE DIMENSIONLESS PARAMETERS
C                 GENERATE VELOCITY LEVELS
C                 GENERATE GRID
C                 GENERATE POTENTIAL DERIVATIVES
C
C
C
C
C
C
C
C
C
C********************************************************************
C
C PURPOSE
C     TO GENERATE SCALING FACTORS WITH RESPECT TO TEMP REF AND DENI1.
C EXTERNALS
C     NONE
C********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER PERSEC,SECIND,PERPAN,PANIND
      COMMON PSAVE(144,10),CSAVE(144,10),PSECTR(144),CSECTR(144),
     1DELEE(20),DELMAX(144),EMAX(144),ETA(20),BSCAT1(144),
     2BSCAT2(144),BSCAT3(144),PHO(20,40),
     1ANG(40),CPHOMX(144),SUNANG,ANGO,BOUNDL(20,40),BOUNDI(20,40),
     2ANGL(20,40),VII(20,40),ICHECK(20,40),RATIO(13),IROOT(144),
     3IFAULT(14),JLOCAL(144),ICON(144),
     1VI(20),EV(8),NLEV,NTIMES,ILEV,TIMIN(20),TIMAX(20),VINUM(20),
     2A(144),B(144),C(144),BB(144),BW(144),TW(144),DW(144),
     3EW(144),TWOCOS(700),
     1IFLG,NPOINT,H,IGEOM,ISYM,IPOIS,NSECT,NPEROD,MPEROD,IDIMY,
     2GAMMA,DVDX(70,48),DVDY(70,48),Y(144),DELTAX,DELTAY,
     3YS(66,48),XS(66),XSS(70),
     1VARBLE(4),T,DELT,SBOUND,NDIM,POT,H,J1,RPI,RPI2,RPIHAF,SQTPI,
     2SAY,TEMPE1,TEMPE2,DENE1,DENE2,DENI1,DENI2,TEMPI1,TEMPI2,
     3TEMPR,LUN3,LUN4,LUN5,LUN6,LUN7,LUN8,LUN9,LUN10,
     1IDTL(144),JDTL(144),PERSEC(144),SECIND(144),IPANEL(144),
     2PERPAN(144,4),PANIND(144),
     1CION(144),CELEC(144),CSEC(144),CSEC1(144),CPHOTO(144),
     2CPHOT1(144),CBSCAT(144),CBSCAI(144),SECPRN(144),SECPRI(144),
     3PHI(144),COUN(144),
     1KASE,JOHL,DPHI,NANG,RAD,PBOUND,OMEGA,ALPHA,SO,JP,IS,ECS,
     2EXPVI2(20),EFTVI(20),IERROR,TETMIN,TETMAX,BETA1,BETA2,JLESS,
     3L2,CARRIV,CPH,XI,CSECEL,CSECEI,CS,CB,CS1,CB1
      DATA PMASSE,CHARGE,BOLTZ/9.0E-31,1.6E-19,1.38E-23/
      DATA TEMPS,TEMPP,TEMPSF,PMASSI/3.0,1.5,30.0,1836.0/
      TEMPR=0.5*TEMPE2
      TEMPR=1000.0
      RATIO(1)=TEMPI2/TEMPR
      RATIO(2)=-TEMPE1/TEMPR
      RATIO(3)=-TEMPS/TEMPR
       RATIO(4)=-TEMPR/TEMPR
      RATIO(8)=TEMPI1 TEMPR
      RATIO(5)=-TEMPP/TEMPR
      RATIO(6)=SQRT(TEMPE1*PMASSI/TEMPR)
      RATIO(7)=SQRT(TEMPI1/TEMPR)
       RATIO(9)=1.0E-6/(0.25*DENI1*CHARGE*SQRT((8.0*TEMPR*CHARGE)/
     1 (RPI*PMASSI*PMASSE)))
      RATIO(10)=-TEMPE2/TEMPR
      RATIO(11)=SQRT(TEMPE2*PMASSI/TEMPR)
      RATIO(12)=SQRT(TEMPI2/TEMPR)
      RATIO(13)=-TEMPSF/TEMPR
      RETURN
      END
      SUBROUTINE GENVEL
C**********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER PERSEC,SECIND,PERPAN,PANIND
      COMMON PSAVE(144,10),CSAVE(144,10),PSECTR(144),CSECTR(144),
     1DELEE(20),DELMAX(144),EMAX(144),ETA(20),BSCAT1(144),
     2BSCAT2(144),BSCAT3(144),PHO(20,40),
     1ANG(40),CPHOMX(144),SUNANG,ANGO,BOUNDL(20,40),BOUNDI(20,40),
     2ANGL(20,40),VII(20,40),ICHECK(20,40),RATIO(13),IROOT(144),
     3IFAULT(14),JLOCAL(144),ICON(144),
     1VI(20),EV(8),NLEV,NTIMES,ILEV,TIMIN(20),TIMAX(20),VINUM(20),
     2A(144),B(144),C(144),BB(144),BW(144),TW(144),DW(144),
     3EW(144),TWOCOS(700),
     1IFLG,NPOINT,H,IGEOM,ISYM,IPOIS,NSECT,NPEROD,MPEROD,IDIMY,
     2GAMMA,DVDX(70,48),DVDY(70,48),Y(144),DELTAX,DELTAY,
     3YS(66,48),XS(66),XSS(70),
     1VARBLE(4),T,DELT,SBOUND,NDIM,POT,H,J1,RPI,RPI2,RPIHAF,SQTPI,
     2SAY,TEMPE1,TEMPE2,DENE1,DENE2,DENI1,DENI2,TEMPI1,TEMPI2,
     3TEMPR,LUN3,LUN4,LUN5,LUN6,LUN7,LUN8,LUN9,LUN10,
     1IDTL(144),JDTL(144),PERSEC(144),SECIND(144),IPANEL(144),
     2PERPAN(144,4),PANIND(144),
     1CION(144),CELEC(144),CSEC(144),CSEC1(144),CPHOTO(144),
     2CPHOT1(144),CBSCAT(144),CBSCAI(144),SECPRN(144),SECPRI(144),
     3PHI(144),COUN(144),
     1KASE,JOHL,DPHI,NANG,RAD,PBOUND,OMEGA,ALPHA,SO,JP,IS,ECS,
     2EXPVI2(20),EFTVI(20),IERROR,TETMIN,TETMAX,BETA1,BETA2,JLESS,
     3L2,CARRIV,CPH,XI,CSECEL,CSECEI,CS,CB,CS1,CB1
C THIS SUBROUTINE GENERATES VELOCITY LEVELS USING THE MONOENERGETIC
C       VELOCITY LEVELS AS A BASE
C NSAMP IS NUMBER OF MONOENERGETIC LEVELS USED
C NLEV=(NSAMP+1)*NTIMES -1 ARE NUMBER OF LEVELS USED
C NTIMES IS AN INTEGER .GE. 1
C VI IS VELOCITY
C**********************************************************************
      NSAMP=8
      EV(1)=1.218128E-01
      EV(2)=3.882449E-01
      EV(3)=7.651497E-01
      EV(4)=1.224691
      EV(5)=1.751398
      EV(6)=2.343383
      EV(7)=3.016609
      EV(8)=3.831371
      NLEV=NSAMP*NTIMES
      VZERO=0.
      DO 100 I=1,NSAMP
      DO 200 J=1,NTIMES
      J1=NTIMES*(I-1)+J
      VI(J1)=VZERO+FLOAT(J)*(EV(I)-VZERO)/FLOAT(NTIMES)
  200 CONTINUE
      VZERO=EV(I)
  100 CONTINUE
C
C STORE FOLLOWING VALUES FOR S/P FIT
      DO 300 K=1,NLEV
      EXPVI2(K)=EXP(-VI(K)**2)
      EFTVI(K)=EFT(VI(K))
  300 CONTINUE
C
C
      RETURN
      END
      SUBROUTINE GENORD(N,PHI,H,RAD,SBOUND,PBOUND,DELTAX,DELTAY,Y,M,
     1 ANGO,IPOIS,PSECTR,CSECTR)
C**********************************************************************
C PURPOSE
C        TO GENERATE GRID POINTS ARRAYS AND COEFFICIENTS OF THE
C        POISSON EQUATION, TO CALL THE GUESS FIELD.
C
C DESCRIPTION OF THE VARIABLES
C       A,B,C ARE COEFFICIENTS IN THE POISSON EQUATION.
C       ANGO IS ANGLE BETWEEN SUN DIRECTION AND ORIGIN OF ANGLES.
C       DELTAX IS GRID SPACING IN X (R) DIRECTION.
C       DELTAY IS GRID SPACING IN Y (ANGLE) DIRECTION
C      NCONS IS USED IN THE COMPUTATION OF H
C      H INITIAL STEP SIZE FOR ORBIT INTEGRATION
C       IPOIS IS A FLAG
C          IPOIS =0    THE GUESS FIELD IS COMPUTED
C          IPOIS=1     COMPUTATION OF THE POISSON EQ
C          IPOIS=2     READ IN POTENTIAL
C**********************************************************************
C
C
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER PERSEC,SECIND,PERPAN,PANIND
      COMMON PSAVE(144,10),CSAVE(144,10),PSECTR(144),CSECTR(144),
     1DELEE(20),DELMAX(144),EMAX(144),ETA(20),BSCAT1(144),
     2BSCAT2(144),BSCAT3(144),PHO(20,40),
     1ANG(40),CPHOMX(144),SUNANG,ANGO,BOUNDL(20,40),BOUNDI(20,40),
     2ANGL(20,40),VII(20,40),ICHECK(20,40),RATIO(13),IROOT(144),
     3IFAULT(14),JLOCAL(144),ICON(144),
     1VI(20),EV(8),NLEV,NTIMES,ILEV,TIMIN(20),TIMAX(20),VINUM(20),
     2A(144),B(144),C(144),BB(144),BW(144),TW(144),DW(144),
     3EW(144),TWOCOS(700),
     1IFLG,NPOINT,H,IGEOM,ISYM,IPOIS,NSECT,NPEROD,MPEROD,IDIMY,
     2GAMMA,DVDX(70,48),DVDY(70,48),Y(144),DELTAX,DELTAY,
     3YS(66,48),XS(66),XSS(70),
     1VARBLE(4),T,DELT,SBOUND,NDIM,POT,H,J1,RPI,RPI2,RPIHAF,SQTPI,
     2SAY,TEMPE1,TEMPE2,DENE1,DENE2,DENI1,DENI2,TEMPI1,TEMPI2,
     3TEMPR,LUN3,LUN4,LUN5,LUN6,LUN7,LUN8,LUN9,LUN10,
     1IDTL(144),JDTL(144),PERSEC(144),SECIND(144),IPANEL(144),
     2PERPAN(144,4),PANIND(144),
     1CION(144),CELEC(144),CSEC(144),CSEC1(144),CPHOTO(144),
     2CPHOT1(144),CBSCAT(144),CBSCAI(144),SECPRN(144),SECPRI(144),
     3PHI(144),COUN(144),
     1KASE,JOHL,DPHI,NANG,RAD,PBOUND,OMEGA,ALPHA,SO,JP,IS,ECS,
     2EXPVI2(20),EFTVI(20),IERROR,TETMIN,TETMAX,BETA1,BETA2,JLESS,
```

```
     3L2,CARRIV,CPH,XI,CSECEL,CSECE1,CS,CB,CS1,CB1
      DATA HCONS/10.0/
C
C
C
      N=NPOINT
      IF(IFLG.EQ.1)GO TO 100
C
C COMPUTE LINEAR SPACE-CHARGE
C
      RLMDA=0.0
      DEBYE=69*SQRT(TEMPR*1.6E-19/(DENI1*1.38E-23))
      IF(IFAULT(14).EQ.1)
     1 RLMDA=(RAD/DEBYE)**2*(DENE1/TEMPE1+DENE2/TEMPE2+DENI1/TEMPI1+
     1 DENI2/TEMPI2)*TEMPR/DENI1
      DELTAX=SBOUND/FLOAT(M+1)
      DELTAY=RPI2/FLOAT(N)
      SYX=(DELTAY/DELTAX)**2
C GENERATE GRID POINTS, COEFFICIENTS AND RHS OF EQUATION (1,NE,1)
  100 MM1=M-1
      DO 200 J=1,N
      Y(J)=FLOAT(J-1)*DELTAY+ANGO
      IF(Y(J).LT.-0.0.OR.Y(J).GT.RPI2) STOP
      YS(M,J)=-SYX*PBOUND
      DO 210 I=2,MM1
      YS(I,J)=0.
  210 CONTINUE
  200 CONTINUE
C
C PREPARE DATA FOR POISSON SOLVER
      TLMDA=RLMDA*DELTAY**2
      DO 220 I=1,M
      XS(I)=FLOAT(I)*DELTAX
      A(I)=SYX
      B(I)=-2.*SYX-TLMDA*EXP(2.*XS(I))
      C(I)=SYX
  220 CONTINUE
      A(1)=0.
      C(M)=0.
C
C
  250 IF(IPOIS-1) 300,301,310
C
C CALL GUESS FIELD
C
  300 CALL GUESS
      IPOIS=IPOIS+1
  301 DO 230 J=1,N
      YS(1,J)=-SYX*PHI(J)
  230 CONTINUE
C
C CALL POISSON SOLVER
      CALL POI
      IF(IERROR.NE.0) WRITE(LUN9,121) IERROR
  121 FORMAT(* ERROR IN POISSON FIELD,IEEROR=*,I3//////)
C
C COMPUTE APPROXIMATE STEPSIZE FOR ORBIT INTEGRATION
      H=DELTAX/HCONS
      IFLG=1
      RETURN
C
C
C
C READ IN POTENTIAL AT THE SURFACE
C
  310 WRITE(LUN9,2345)
 2345 FORMAT (/////20X,*THE INITIAL POTENTIAL AT THE SURFACE IS*,
     1 /20X*-----------------------------------------*//)
C
      READ(LUN10,122) (PHI(J),J=1,N)
  122 FORMAT(6E12.5)
      WRITE (LUN9,1220) (PHI(J),J=1,N)
 1220 FORMAT(6(1X,1PE12.5))
      IPOIS=1
      GO TO 301
C
      END
      SUBROUTINE DIVPOT
C
C*******************************************************************
C
C PURPOSE
C TO COMPUTE DERIVATIVES OF POTENTIAL IN X AND Y DIRECTIONS AND
C TO RESET POTENTIAL ARRAY.
C DESCRIPTION OF THE VARIABLES
C       DVDX IS N.D. FINITE DIFFERENCE DERIVATIVBE OF POTENTIAL IN X DIF
C       DVDY IS N.D. FINITE DIFFERENCE DERIVATIVE OF POTENTIAL IN Y DIR
C       N IS NUMBER OF INTERVALS IN Y DIRECTION
C       MP2 IS NUMBER OF GRID POINTS IN X DIRECTION
C       PHI IS POTENTIAL AT THE SUFACE OF SPACECRAFT
C       PBOUND IS OUTER BOUNDARY FOR POTENTIAL
C       SBOUND IS OUTER LIMIT FOR RADIAL DISTANCE
C       XS IS RADIAL DISTANCE IN I DIRECTION INCLUDING BOUNDARY POINTS
C       XSS IS ARRAY USED IN FUNCTION RUNGE
C       YS IS POTENTIAL MATRIX CHANGED TO INCLUDE BOUNDARY POINTS
C
C EXTERNALS   NONE
C*******************************************************************
      INTEGER PERSEC,SECIND,PERPAN,PANIND
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON PSAVE(144,10),CSAVE(144,10),PSECTR(144),CSECTR(144),
     1DELEE(20),DELMAX(144),EMAX(144),ETA(20),BSCAT1(144),
     2BSCAT2(144),BSCAT3(144),PHO(20,40),
     1ANG(40),CPHOMX(144),SUMANG,ANGO,BOUNDL(20,40),BOUNDI(20,40),
     2ANGL(20,40),VII(20,40),ICHECK(20,40),RATIO(13),IROOT(144),
     3IFAULT(14),JLOCAL(144),ICON(144),
     1VI(20),EV(8),NLEV,NTIMES,ILEV,TIMIN(20),TIMAX(20),VIMUM(20),
     2A(144),B(144),C(144),BB(144),BW(144),TW(144),BW(144),
     3EW(144),TWOCOS(700),
     1IFLG,NPOINT,M,IGEOM,ISYM,IPOIS,NSECT,NPEROD,MPEROD,IDINY,
     2GAMMA,DVDX(70,48),DVDY(70,48),Y(144),DELTAX,DELTAY,
     3YS(66,48),XS(66),XSS(70),
     1VARBLE(4),T,DELT,SBOUND,NDIM,POT,H,J1,RPI,RPI2,RPIHAF,SQTPI,
     2SAY,TEMPE1,TEMPE2,DENE1,DENE2,DENI1,DENI2,TEMPI1,TEMPI2,
     3TEMPR,LUN3,LUN4,LUN5,LUN6,LUN7,LUN8,LUN9,LUN10,
     1IDTL(144),JDTL(144),PERSEC(144),SECIND(144),IPANEL(144),
     2PERPAN(144,4),PANIND(144),
     1CION(144),CELEC(144),CSEC(144),CSEC1(144),CPHOTO(144),
     2CPHOT1(144),CBSCAT(144),CBSCA1(144),SECPRN(144),SECPRI(144),
     3PHI(144),COUN(144),
     1KASE,JOML,DPHI,NANG,RAD,PBOUND,OMEGA,ALPHA,SO,JP,IS,ECS,
     2EXPVI2(20),EFTVI(20),IERROR,TETMIN,TETMAX,BETA1,BETA2,JLESS,
     3L2,CARRIV,CPH,XI,CSECEL,CSECE1,CS,CB,CS1,CB1
C
      MP3=M+3
      MP2=M+2
      MP1=M+1
      N=NPOINT
C
      XS(MP2)=SBOUND
      DO 10 I=2,MP1
      IPA=MP2-I
      IPB=IPA+1
      XS(IPB)=XS(IPA)
   10 CONTINUE
      XS(1)=SO
      XSS(1)=SO
      XSS(2)=SO+0.5*DELTAX
      DO 20 I=3,MP2
        XSS(I)=XSS(I-1)+DELTAX
   20 CONTINUE
      XSS(MP3)=SBOUND
C
      DO 40 J=1,N
      YS(MP2,J)=PBOUND
      DO 30 I=2,MP1
      IPA=MP2-I
      IPB=IPA+1
      YS(IPB,J)=YS(IPA,J)
   30 CONTINUE
      YS(1,J)=PHI(J)
   40 CONTINUE
C
C COMPUTATION OF THE DERIVATIVES IN RADIAL DISTANCES
C BY FINITE DIFFERENCES
      DO 50 J=1,N
      DVDX(1,J)=(-YS(3,J)+4.0*YS(2,J)-3.0*YS(1,J))/(2.0*DELTAX)
      DVDX(MP3,J)=(3.0*YS(MP2,J)-4.0*YS(MP1,J)+YS(M,J))/(2.0*DELTAX)
   50 CONTINUE
C
      DO 60 J=1,N
      DO 60 I=2,MP2
      DVDX(I,J)=(YS(I,J)-YS(I-1,J))/DELTAX
   60 CONTINUE
C
C CALCULATION OF DVDY(IN J DIRECTION OR ANGLES)
      DO 70 I=1,MP2
      DO 70 J=1,N
      JP1=J+1
      IF (J.EQ.N) JP1=1
      DVDY(I,J)=(YS(I,JP1)-YS(I,J))/DELTAY
   70 CONTINUE
      RETURN
      END
      SUBROUTINE ION
C
C
C
C
C
C
C
C
C                         PACKAGE # 3
C                    ---------------------
C
C CURRENT COLLECTION OF PRIMARY PARTICLES.COMPUTE ENERGINO SEC CURRENTS.
C
C
C
C
C
C
C
C
C
C
C***************************************************************L
C PURPOSE
C
C TO COMPUTE ION CURRENT AT A GIVEN GRID POINT. THE DISTRIBUTION FUNCTIO
C IS APPROXIMATED BY A DISCRETE NUMBER OF LEVELS. FOR EACH OF THESE
C  LEVELS, THE PHASE SPACE BOUNDARIES ARE DETERMINED BY FOLLOWING
C ION TRAJECTORIES BACKWARDS.
C I=SUM FROM J=1 TO J=JMAX OF INTEGRAL FROM U(I) TO U(I+1) OF
C DU*F(U)*U*U*(COS(ANG1(U))-COS(ANG2(U))
C
C DESCRIPTION OF THE VARIABLES
C BETA1=VI**2 IS NONDIMENSIONAL ENERGY OF IONS AT INFINITY
C BETA2=BETA1-PHI(J) IS ION KINETIC ENERGY AT SURFACE.
C CION IS ION CURRENT
C DELP COEFFICIENT FOR SECONDARY EMISSION DUE TO PROTONS
C EXMP COEFFICIENT FOR SECONDARY EMISSION DUE TO PROTONS
C N IS NUMBER OF GRID POINT AT SURFACE
C NLEV IS TOTAL NUMBER OF MONOENERGETIC LEVELS
C ILEV SELECTS THE NUMBER OF MONOENERGETIC LEVELS USED (ILEV=2
C ITER NUMBER OF ITERATIONS
C JOML IS FLAG
C       IF JOML=0,ORBITS ARE FOLLOWED NUMERICALLY.
C       IF JOML=1,OML CONDITIONS ARE ASSUMED.
C SECPRN CURRENT DUE TO SECONDARY ELECTRONS INDUCED BY PROTONS
C SECPR1 CURRENT DUE TO SECONDARY ELECTRONS INDUCED BY PROTONS
C LEAVING THE SURFACE.
C TETMIN MAXIMUM ACCECPTANCE ANGLE
C TETMIN MINIMUM ACCEPTANCE ANGLE
C VACC IS ACCURACY DESIRED ON VELOCTY SPACE MINIMUM
C VZERO IS MINIMUM KINETIC VELOCITY AT SPACECRAFT SURFACE.
C
C
C EXTERNALS
C       AMAX1
C       COEFT
C        EXP
C       GENORD
C        ORBIT
C
C*******************************************************************
      INTEGER PERSEC,SECIND,PERPAN,PANIND
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON PSAVE(144,10),CSAVE(144,10),PSECTR(144),CSECTR(144),
     1DELEE(20),DELMAX(144),EMAX(144),ETA(20),BSCAT1(144),
     2BSCAT2(144),BSCAT3(144),PHO(20,40),
     1ANG(40),CPHOMX(144),SUMANG,ANGO,BOUNDL(20,40),BOUNDI(20,40),
     2ANGL(20,40),VII(20,40),ICHECK(20,40),RATIO(13),IROOT(144),
     3IFAULT(14),JLOCAL(144),ICON(144),
     1VI(20),EV(8),NLEV,NTIMES,ILEV,TIMIN(20),TIMAX(20),VIMUM(20),
     2A(144),B(144),C(144),BB(144),BW(144),TW(144),BW(144),
     3EW(144),TWOCOS(700),
     1IFLG,NPOINT,M,IGEOM,ISYM,IPOIS,NSECT,NPEROD,MPEROD,IDINY,
     2GAMMA,DVDX(70,48),DVDY(70,48),Y(144),DELTAX,DELTAY,
     3YS(66,48),XS(66),XSS(70),
     1VARBLE(4),T,DELT,SBOUND,NDIM,POT,H,J1,RPI,RPI2,RPIHAF,SQTPI,
     2SAY,TEMPE1,TEMPE2,DENE1,DENE2,DENI1,DENI2,TEMPI1,TEMPI2,
     3TEMPR,LUN3,LUN4,LUN5,LUN6,LUN7,LUN8,LUN9,LUN10,
     1IDTL(144),JDTL(144),PERSEC(144),SECIND(144),IPANEL(144),
     2PERPAN(144,4),PANIND(144),
     1CION(144),CELEC(144),CSEC(144),CSEC1(144),CPHOTO(144),
     2CPHOT1(144),CBSCAT(144),CBSCA1(144),SECPRN(144),SECPRI(144),
     3PHI(144),COUN(144),
     1KASE,JOML,DPHI,NANG,RAD,PBOUND,OMEGA,ALPHA,SO,JP,IS,ECS,
     2EXPVI2(20),EFTVI(20),IERROR,TETMIN,TETMAX,BETA1,BETA2,JLESS,
     3L2,CARRIV,CPH,XI,CSECEL,CSECE1,CS,CB,CS1,CB1
      DATA VACC/1.0E-3/
C*******************************************************************
```

```
[illegible]
      IF(JOML.EQ.0) GO TO 714
C COMPUTE ION CURRENT COLLECTION ASSUMING OML CONDITIONS
      CSECEL=0.0
      CSECE1=0.0
      CI1=0.0
      CI2=0.0
      XI=PHI(J)
      IF(IFAULT(9).EQ.1) CALL SECEL
      SECPRN(J)=CSECEL
      SECPRI(J)=CSECE1
      IF(XI.GT.0.0) GO TO 713
      IF(TEMPI1.NE.0.0) CI1=(2.0*SAY*(SQRT(-XI/RATIO(8))+
     1 COEFT(SQRT(-XI/RATIO(8)))))*RATIO(7)
      IF(TEMPI2.NE.0.) CI2=(2.0*SAY*(SQRT(-XI/RATIO(1))+
     1 COEFT(SQRT(-XI/RATIO(1)))))*RATIO(12)*DENI2/DENI1
      CION(J)=CI1+CI2
      GO TO 715
  713 IF (TEMPI1.NE.0.0) CI1=EXP(-XI/RATIO(8))*RATIO(7)
      IF(TEMPI2.NE.0.) CI2=EXP(-XI/RATIO(1))*RATIO(12)*DENI2/DENI1
      CION(J)=CI1+CI2
      GO TO 715
  714 CONTINUE
C*********************************************************************D+482C
C
C
C LOOP IF BIMAXWELLIAN DISTRIBUTION
      NDIM=4
      DO 710 JJ=1,2
      IF (JJ.NE.1) GO TO 711
      GAMMA=RATIO(8)
      TEMPI=TEMPI1
      DEN=1.0
      RATE=RATIO(7)
      GO TO 712

  711 CONTINUE
      GAMMA=RATIO(1)
      TEMPI=TEMPI2
      DEN=DENI2/DENI1
      RATE=RATIO(12)
  712 CONTINUE
C
C
      NUM=1
      IF(DEN.EQ.0.0) GO TO 710
      NTOUR=1
      IF (PHI(J).GT.0.0) NUM=1
C C COMPUTE MINIMUM KINETIC VELOCITY AT COLLECTOR SURFACE.
      VZERO=0.0
      IF(PHI(J).NE.0.0)
     1VZERO=AMAX1(0.0,-PHI(J)/GAMMA*SQRT(ABS(PHI(J)/GAMMA))/
     2 ABS(PHI(J)/GAMMA))
C
C COMPUTE FOR WHICH ENERGY LEVELS,PARTICLES ARE ESCAPING.
      DO 700 I=1,NLEV,ILEV
      BETA1=VI(I)**2
      BETA2=BETA1-PHI(J)/GAMMA
      IF(BETA2.LE.0.)GO TO 700
C IF ENERGY AT SURFACE(BETA2) IS LESS THAN ZERO, GO ON TO NEXT ENERGY
      TETMIN=0.0
      TETMAX=RPI
C COMPUTE ACCEPTANCE ANGLES FOR A GIVEN ENERGY LEVEL
C IF ORBITS CORRESPONDING TO LOW ENERGY LEVELS ARE OML,DO NOT
C COMPUTE ORBITS FOR HIGHER ENRGY LEVELS.
      IF (NTOUR.EQ.1)
     1 CALL GENORB
      IF(JLESS.EQ.2)GO TO 700
      IF(TETMIN.LE.0.0.AND.TETMAX.GE.RPI) NTOUR=0
      NUM=NUM+1
      TIMIN(NUM)=TETMIN
      TIMAX(NUM)=TETMAX
      VINUM(NUM)=SQRT(BETA2)
C COMPUTE COEFFICIENT FOR SECONDARY ELECTRONS INDUCED BY PROTONS
      IF(IFAULT(9).EQ.1)
     1 DELEE(NUM)=DELP(J)*SQRT(0.001*BETA2*TEMPI)/(1.0+BETA2*TEMPI/
     1 EXMP(J))
  700 CONTINUE
C
      IF(NUM.LE.1)GO TO 705
C
C
C COMPUTE ACCEPTANCE ANGLES FOR THE LOWEST ENERGY LEVEL
      TZERO=(TIMIN(2)+TIMAX(2))/2.
      VMIN=VINUM(2)
  810 VMED=(VMIN+VZERO)/2.
      VARBLE(1)=SO
      VARBLE(2)=VMED*SIN(TZERO)
      VARBLE(3)=Y(J)
      VARBLE(4)=-VMED*COS(TZERO)
      T=0.
      DELT=H
      BETA1=VMED*VMED+PHI(J)/GAMMA
      CALL ORBIT
      IF(VARBLE(1).GE.(SBOUND-DELTAX))VMIN=VMED
      IF(VARBLE(1).LT.SO)VZERO=VMED
      IF(ABS(VMIN-VZERO).GT.VACC)GO TO 810
      VINUM(1)=VMIN
      BETA2=VMIN**2
      BETA1=BETA2+PHI(J)/GAMMA
      TETMIN=TIMIN(2)
      TETMAX=TIMAX(2)
      TIMIN(1)=TZERO
      TIMAX(1)=TZERO
      CALL GENORB
      IF(JLESS.EQ.2) GO TO 800
      TIMIN(1)=TETMIN
      TIMAX(1)=TETMAX
  800 NUMM1=NUM-1
      IF(IFAULT(9).EQ.1)
     1 DELEE(1)=DELP(J)*SQRT(0.001*BETA2*TEMPI)/(1.0+BETA2*TEMPI/
     1 EXMP(J))
      WRITE(LUN6,610)J,JJ,(TIMIN(I),I=1,NUM)
  610 FORMAT (1X,*J=*,I5,*JJ=*,I5,*ACCEPTANCE ANGLES FOR IONS*/
     1 (10(1X,1PE10.3)))
      WRITE (LUN6,611) (TIMAX(I),I=1,NUM)
  611 FORMAT (10(1X,1PE10.3))
      WRITE (LUN6,612)
  612 FORMAT (//1X,*VELOCITY LEVELS FOR IONS*/)
      WRITE (LUN6,611) (VINUM(K),K=1,NUM)
C
C
C
C
C START INTEGRATION
C
C
      CUR1=0.
      CUR2=0.0
      COEF1=EXP(-PHI(J)/GAMMA)/SQTPI
      AMIX0=0.
      BMIX0=0.
      ADEL0=0.0
      BDEL0=0.0
      DO 816 I=1,NUMM1
      DENOM=VINUM(I+1)-VINUM(I)
      AMIN=(COS(TIMIN(I+1))-COS(TIMIN(I)))/DENOM
      BMIN=COS(TIMIN(I))-AMIN*VINUM(I)
      AMAX=(COS(TIMAX(I+1))-COS(TIMAX(I)))/DENOM
      BMAX=COS(TIMAX(I))-AMAX*VINUM(I)
      AMIX1=AMIN-AMAX
      BMIX1=BMIN-BMAX
      CUR1=CUR1+EXP(-VINUM(I)**2)*((AMIX1-AMIX0)*(1.+VINUM(I)**2)+
     1(BMIX1-BMIX0)*(VINUM(I)+COEFT(VINUM(I))))
      IF (IFAULT(9).EQ.0.0) GO TO 823
      ADEL1=(COS(TIMIN(I+1))*DELEE(I+1)-COS(TIMIN(I))*DELEE(I))/DENOM
      BDEL1=DELEE(I)*COS(TIMIN(I))-ADEL1*VINUM(I)
      ADELM=(COS(TIMAX(I+1))*DELEE(I+1)-COS(TIMAX(I))*DELEE(I))/DENOM
      BDELM=DELEE(I)*COS(TIMAX(I))-ADELM*VINUM(I)
      ADEL1=ADEL1-ADELM
      BDEL1=BDEL1-BDELM
      CUR2=CUR2+EXP(-VINUM(I)**2)*((ADEL1-ADEL0)*(1.0+VINUM(I)**2)+
     1 (BDEL1-BDEL0)*(VINUM(I)+COEFT(VINUM(I))))
      ADEL0=ADEL1
      BDEL0=BDEL1
  823 AMIX0=AMIX1
      BMIX0=BMIX1
  816 CONTINUE
C
C COMPUTATION OF ION CURRENT
C
      CION(J)=CION(J)+COEF1*CUR1*RATE*DEN
      CUR22=CUR2
      IF (PHI(J).GT.0.) CUR2=
     1 (2.0*SAY*(SQRT(-PHI(J)/RATIO(13))+COEFT(SQRT(-PHI(J)/
     2 RATIO(13)))))*EXP(PHI(J)/RATIO(13))*CUR2
      SECPRN(J)=SECPRN(J)+COEF1*CUR2*RATE*DEN
      SECPRI(J)=SECPRI(J)+COEF1*CUR22*RATE*DEN
      GO TO 710
  705 CONTINUE
  710 CONTINUE
  715 CONTINUE
      RETURN
      END
      SUBROUTINE ELEC
C*********************************************************************
C PURPOSE
C          TO COMPUTE ELECTRON CURRENT AT EACH GRID POINT AND
C CURRENT DUE TO EMITTED SECONDARIES AND BACKSCATTERS
C I=SUM FROM J=1 TO J=JMAX OF INTEGRAL FROM U(I) TO U(I+1)
C OF DU*F(U)*U*U*(COS(ALPHA2(U)-COS(ALPHA1(U))
C
C DESCRIPTION OF THE VARIABLES
C        BETA1=VI**2 IS NONDIMENSIONAL ELECTRON ENERGY.
C        BETA2=BETA1-PHI(J)/GAMMA IS ELECTRON KINETIC ENERGY AT SURFACE
C        CSEC IS SECONDARY CURRENT
C        CSEC1 IS SECONDARY CURRENT LEAVING THE SURFACE
C        CBSCAT IS BSCATTER CURRENT
C        CBSCA1 IS BSCATTER CURRENT LEAVING THE SURFACE
C        ETA IS COEFFICIENT USED IN THE COMPUTATION OF BSCATTERS.
C        DELEE IS COEFFICIENT USED IN THE COMPUTATION OF SECONDARIES
C        ,THE COEFFICIENTS IN THE FORMALUE CORRESPOND TO STERNGLASS
C        EQUATION.THEY ARE TAKEN FROM KNOTT'S PAPER.
C        ILEV SELECTS THE NUMBER OF MONOENERGETIC LEVELS BEFORE
C        ITER IS NEMBER OF ITERATIONS
C        IFAULT IS A FLAG FOR SECONDARIES AND BASCATTERS.
C JOML IS A FLAG
C       JOML=0,ACCEPTANCE ANGLES ARE COMPUTED NUMERICALLY.
C       JOML=1,ACCEPTANCE ANGLES ARE ASSUMED TO BE OML
C        TETMAX IS MAXIMUM ACCEPTANCE ANGLE.
C        TETMIN IS MINIMUM ACCECPTANCE ANGLE.
C        VACC IS ACCURACY DESIRED ON VELOCITY SPACE MINI
C        VZERO IS MINIMUM KINETIC VELOCITY AT COLLECTOR SURFACE
C
C
C EXTERNALS
C        AMAX1
C        COEFT
C        EXP
C        GENORB
C        ORBIT
C*********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER PERSEC,SECIND,PERPAN,PANIND
      COMMON PSAVE(144,10),CSAVE(144,10),PSECTR(144),CSECTR(144),
     1DELEE(20),DELMAX(144),EMAX(144),ETA(20),BSCAT1(144),
     2BSCAT2(144),BSCAT3(144),PHO(20,40),
     1ANG(40),CPHONX(144),BUNANG,ANGD,BOUNBL(20,40),BOUNDI(20,40),
     2ANGL(20,40),VII(20,40),ICHECK(20,40),RATIO(13),IROOT(144),
     3IFAULT(14),JLOCAL(144),ICON(144),
     1VI(20),EV(8),NLEV,NTIMES,ILEV,TIMIN(20),TIMAX(20),VINUM(20),
     2A(144),B(144),C(144),BB(144),BW(144),TW(144),DW(144),
     3EW(144),TWOCOS(700),
     1IFLO,NPOINT,N,IOEOM,IBYM,IPOIS,NSECT,NPEROD,MPEROD,IDIMY,
     2GAMMA,DVDX(70,40),DVDY(70,40),Y(144),DELTAX,DELTAY,
     3YS(66,40),XS(66),XSS(70),
     1VARBLE(4),T,DELT,SBOUND,NDIM,POT,H,J1,RPI,RPI2,RPINAF,SQTPI,
     2SAY,TEMPE1,TEMPE2,DENE1,DENE2,DENI1,DENI2,TEMPI1,TEMPI2,
     3TEMPR,LUN3,LUN4,LUN5,LUN6,LUN7,LUN8,LUN9,LUN10,
     1IDTL(144),JDTL(144),PERSEC(144),SECIND(144),IPANEL(144),
     2PERPAN(144,4),PANIND(144),
     1CION(144),CELEC(144),CSEC(144),CSEC1(144),CPHOTO(144),
     2CPHOT1(144),CBSCAT(144),CBSCA1(144),SECPRN(144),SECPRI(144),
     3PHI(144),COUN(144),
     1BASE,JOML,DPHI,NANG,RAD,PBOUND,OMEGA,ALPHA,SO,JP,IS,ECS,
     2EXPVI2(20),EFTVI(20),IERROR,TETMIN,TETMAX,BETA1,BETA2,JLESS,
     3L2,CARRIV,CPH,XI,CSECEL,CSECE1,CS,CB,CS1,CB1
      DATA VACC/1.0E-3/
C*********************************************************************
      J=JP
C
C INITIALIZE CURRENTS TO ZERO
      NDIM=4
      CELEC(J)=0.0
      CSEC(J)=0.0
      CBSCAT(J)=0.0
      CSEC1(J)=0.0
      CBSCA1(J)=0.0
      IF(JOML.EQ.0) GO TO 714
C
C COMPUTE ELECTRON CURRENT USING OML CONDITIONS
      XI=PHI(J)
      CS=0.0
      CB=0.0
      CS1=0.0
      CB1=0.0
      CE1=0.0
      CE2=0.0
      IF(IFAULT(3).NE.0.0.OR.IFAULT(4).NE.0) CALL CBS
      CSEC(J)=CS
      CSEC1(J)=CS1
```

```
      CBSCAT(I)=CB
      CBSCAT(I)=CBI
      IF(XI.GT.0.0) GO TO 713
      IF(RATIO(2).NE.0.0) CE1=RATIO(6)*EXP( XI/RATIO(2))
      IF(RATIO(11).NE.0.0) CE2=RATIO(11)*EXP( XI/RATIO(10))
      CELEC(J)=(CE1*DENE1+CE2*DENE2)/DENI1
      GO TO 715
  713 IF(RATIO(2).NE.0.0) CE1=RATIO(6)*(2.0*SAY*(SORT( XI/RATIO(2))+
     1 COEFT(SORT(-XI/RATIO(2)))))
      IF(RATIO(11).NE.0.0) CE2=RATIO(11)*(2.0*SAY*(SORT(-XI/RATIO(10))
     1 + COEFT(SORT( XI RATIO(10)))))
      CELEC(J)=(CE1*DENE1+CE2*DENE2) DENI1
      GO TO 715
  714 CONTINUE
C
C
C
C PREPARE DATA FOR BIMAXWELLIAN DISTR
      DO 710 JJ=1,2
      IF(JJ.NE.1) GO TO 711
      GAMMA=RATIO(2)
      TEMPE=TEMPE1
      RATE=RATIO(6)
      DEN=DENE1/DENI1
      GO TO 712
  711 GAMMA=RATIO(10)
      TEMPE=TEMPE2
      RATE=RATIO(11)
      DEN=DENE2/DENI1
  712 CONTINUE
      IF(DEN.EQ.0.0) GO TO 710
C COMPUTE ENERGY LEVELS AND ACCEPTANCE ANGLES
      NTOUR=1
      NUM=1
C COMPUTE MINIMAL KINETIC VELOCITY
      VZERO=0.0
      IF(PHI(J).NE.0.0)
     1VZERO=AMAX1(0.,-PHI(J)/GAMMA*SORT(ABS(PHI(J)/GAMMA))/
     1 ABS(PHI(J)/GAMMA))
C
C
      DO 705 I=1,NLEV,ILEV
      BETA1=VI(I)**2
      BETA2=BETA1-PHI(J)/GAMMA
      IF(BETA2.LE.0.)GO TO 705
C IF ENERGY AT SURFACE(BETA2) IS LESS THAN ZERO, GO ON TO NEXT ENERGY
      TETMIN=0.0
      TETMAX=RPI
C COMPUTE ACCEPTANCE ANGLES
C IF ORBITS CORRESPONDING TO LOW ENERGY LEVELS ARE OML, DO NOT COMPUTE
C ORBITS NUMERICALLY FOR HIGHER ENERGY LEVELS.
      IF (NTOUR.EQ.1)
     1 CALL GENORB
      IF(JLESS.EQ.2)GO TO 705
      NUM=NUM+1
      IF (TETMIN.LE.0.0.AND.TETMAX.GE.RPI) NTOUR=0
      TIMIN(NUM)=TETMIN
      TIMAX(NUM)=TETMAX
      VINUM(NUM)=SORT(BETA2)
C COMPUTE COEFFICIENTS FOR SECONDARY ELECTRONS
      IF(IFAULT(3).EQ.1)DELEE(NUM)=7.4*DELMAX(J)*BETA2*TEMPE*EXP(
     1-2.*SQRT(BETA2*TEMPE/EMAX(J)))/EMAX(J)
      IF(IFAULT(4).EQ.0)GO TO 705
C COMPUTE COEFFICIENTS FOR BACKSCATTER ELECTRONS
C NOTE :FOR BSCAT3,THE CORRESPONDING ENERGY IS IN KV
      BSC=BSCAT3(J)*BETA2*GAMMA
      ETA(NUM)=BSCAT1(J)
      IF (BSC.LE.100.)
     1 ETA(NUM)=BSCAT1(J)-BSCAT2(J)*EXP(-BSC)
  705 CONTINUE
      IF(NUM.EQ.1)GO TO 709
C
C COMPUTE ACCEPTANCE ANGLES FOR LOWESTE NERGY LEVEL
C
      TZERO=(TIMIN(2)+TIMAX(2))/2.
      VMIN=VINUM(2)
  817 VMED=(VMIN+VZERO)/2.
      VARBLE(1)=SO
      VARBLE(2)=VMED*SIN(TZERO)
      VARBLE(3)=Y(J)
      VARBLE(4)=-VMED*COS(TZERO)
      T=0.
      DELT=H
      BETA1=VMED*VMED+PHI(J)/GAMMA
      CALL ORBIT
      IF(VARBLE(1).GE.(SBOUND-DELTAX))VMIN=VMED
      IF(VARBLE(1).LT.SO)VZERO=VMED
      IF(ABS(VMIN-VZERO).GT.VACC)GO TO 817
      VINUM(1)=VMIN
      BETA2=VMIN**2
      BETA1=BETA2+PHI(J)/GAMMA
      TETMIN=TIMIN(2)
      TETMAX=TIMAX(2)
      TIMIN(1)=TZERO
      TIMAX(1)=TZERO
      CALL GENORB
      IF(JLESS.EQ.2) GO TO 801
      TIMIN(1)=TETMIN
      TIMAX(1)=TETMAX
  801 CONTINUE
      IF(IFAULT(3).EQ.1)DELEE(1)=7.4*DELMAX(J)*BETA2*TEMPE*EXP(-2.
     1*SQRT(BETA2*TEMPE/EMAX(J)))/EMAX(J)
      IF(IFAULT(4).EQ.0)GO TO 310
      BSC=BSCAT3(J)*BETA2*GAMMA
      ETA(1)=BSCAT1(J)
      IF(BSC.LE.100.0)
     1 ETA(1)=BSCAT1(J)-BSCAT2(J)*EXP(-BSC)
  310 CONTINUE
C VZERO,TZERO ARE VELOCITY,ANGLE CO-ORDINATE MINIMA
      NUMM1=NUM-1
      WRITE (LUN6,615) J,JJ,(TIMIN(I),I=1,NUM)
  615 FORMAT (1X,*J=*,I5,* JJ=*,I5,*ACCEPTANCE ANGLES FOR ELECTRONS*
     1 / (10(1X,1PE10.3)))
      WRITE (LUN6,616) (TIMAX(I),I=1,NUM)
  616 FORMAT (10(1X,1PE10.3))
      WRITE (LUN6,617)
  617 FORMAT (//* VELOCITY LEVELS FOR ELECTRONS*/)
      WRITE (LUN6,616) (VINUM(K),K=1,NUM)
C
C
C
C
C START INTEGRATION
C
C
      CUR1=0.0
      CUR2=0.
      COFF1=EXP(-PHI(J)/GAMMA)/SQTPI
      AMIXO=0.
      BMIXO=0.
      ADELO=0.
      BDELO=0.
      CUR3=0.
      ASCAO=0.
      BSCAO=0.
C
C
C
      DO 816 I=1,NUMM1
      DENOM=VINUM(I+1) VINUM(I)
      AMIN=(COS(TIMIN(I+1))-COS(TIMIN(I)))/DENOM
      BMIN=COS(TIMIN(I)) AMIN*VINUM(I)
      AMAX=(COS(TIMAX(I+1))-COS(TIMAX(I)))/DENOM
      BMAX=COS(TIMAX(I))-AMAX*VINUM(I)
      AMIX1=AMIN-AMAX
      BMIX1=BMIN-BMAX
      CUR1=CUR1+EXP(-VINUM(I)**2)*((AMIX1-AMIXO)*(1.+VINUM(I)**2)+
     1(BMIX1-BMIXO)*(VINUM(I)+COEFT(VINUM(I))))
      IF(IFAULT(3).EQ.0)GO TO 823
      ADELI (COS(TIMIN(I+1))*DELEE(I+1)-COS(TIMIN(I))*DELEE(I))/DENOM
      BDELI=DELEE(I)*COS(TIMIN(I))-ADELI*VINUM(I)
      ADELM=(COS(TIMAX(I+1))*DELEE(I+1)-COS(TIMAX(I))*DELEE(I))/DENOM
      BDELM=DELEE(I)*COS(TIMAX(I))-ADELM*VINUM(I)
      ADEL1=ADELI-ADELM
      BDEL1-BDELI-BDELM
      CUR2=CUR2+EXP(-VINUM(I)**2)*((ADEL1-ADELO)*(1.+VINUM(I)**2)+
     1(BDEL1-BDELO)*(VINUM(I)+COEFT(VINUM(I))))
  823 IF(IFAULT(4).EQ.0)GO TO 824
      ASCAI=(COS(TIMIN(I+1))*ETA(I+1)-COS(TIMIN(I))*ETA(I))/DENOM
      BSCAI=ETA(I)*COS(TIMIN(I))-ASCAI*VINUM(I)
      ASCAM=(COS(TIMAX(I+1))*ETA(I+1)-COS(TIMAX(I))*ETA(I))/DENOM
      BSCAM=ETA(I)*COS(TIMAX(I))-ASCAM*VINUM(I)
      ASCA1=ASCAI-ASCAM
      BSCA1=BSCAI-BSCAM
      CUR3=CUR3+EXP(-VINUM(I)**2)*((ASCA1-ASCAO)*(1.+VINUM(I
     1)**2)+(BSCA1-BSCAO)*(VINUM(I)+COEFT(VINUM(I))))
  824 CONTINUE
  814 AMIXO=AMIX1
      BMIXO=BMIX1
      IF(IFAULT(3).EQ.0)GO TO 813
      ADELO=ADEL1
      BDELO=BDEL1
  813 IF(IFAULT(4).EQ.0)GO TO 816
      ASCAO=ASCA1
      BSCAO=BSCA1
  816 CONTINUE
C
C COMPUTE ELECTRON CURRENT
C
      CUR22=CUR2
      CUR33=CUR3
      CELEC(J)=RATE*COFF1*CUR1*DEN+CELEC(J)
      IF (PHI(J).GT.0.0) CUR2=
     1 (2.0*SAY*(SORT(-PHI(J)/RATIO(3))+COEFT(SORT(-PHI(J)/
     2 RATIO(3)))))*EXP(PHI(J)/RATIO(3))*CUR2
      IF(PHI(J).GT.0.) CUR3=
     1 (2.0*SAY*(SORT(-PHI(J)/RATIO(4))+COEFT(SORT(-PHI(J)/
     2 RATIO(4)))))*EXP(PHI(J)/RATIO(4))*CUR3
      CSEC(J)=RATE*COFF1*CUR2*DEN+CSEC(J)
      CSEC1(J)=CSEC1(J)+RATE*COFF1*CUR22*DEN
      CBSCAT(J)=RATE*COFF1*CUR3*DEN+CBSCAT(J)
      CBSCA1(J)=CBSCA1(J)+RATE*COFF1*CUR33*DEN
      GO TO 710
  709 CONTINUE
  710 CONTINUE
  715 CONTINUE
      RETURN
      END
      FUNCTION COEFT(EX)
      IMPLICIT REAL*8 (A-H,OZ)
C     COEFT(EX) = 0.5 * ROOT OF PI * EXP(EX*EX) * (1.0-EFT(EX))
C     EFT(EX) = 2.0/ROOT OF PI * INTEGRAL FROM 0 TO EX OF EXP(-T*T)*DT
C     SELF-SUFFICIENT, MORE ACCURATE VERSION USING TAYLOR AND ASYMPTOTIC
      IF(EX.LE.3.2) GO TO 10
      VIN=0.5/EX
      V=VIN/EX
      CENT=945.-V*(10395.-V*(135135.-V*(2027025.-V*(34459425.
     1 -V*327364537.5))))
      COEFT  =   (((((-CENT*V+105.0)*V-15.0)*V+3.0)*V-1.0)*V+1.0)*VIN
      RETURN
   10 W=EX*EX
      FNC=0.0
      IF(EX.GT.2.8)FNC=
     J     -W*( 2.483069097454913E-47-W*( 6.205657919637400E-49
     K     -W*( 1.513107949541218E-50-W*( 3.601579309810128E-52
     L     -W*( 8.373419683872287E-54-W*( 1.902541227289881E-55
     M     -W*( 4.226789754193555E-57-W*( 9.186429502398693E-59
     N     -W*( 1.954102582324173E-60-W*( 4.070135277853260E-62
     O     -W*( 8.304614505929117E-64-W*( 1.660580513451091E-65
     P ))) ))) ))) )))
      FNB=0.0
      IF(EX.GT.2.1)FNB=
     D     -W*( 4.678483515518486E-29-W*( 1.669761793417372E-30
     E     -W*( 5.754191643982173E-32-W*( 1.916942862109783E-33
     F     -W*( 6.180307588222798E-35-W*( 1.930357208815108E-36
     G     -W*( 5.846755007468838E-38-W*( 1.710856062801784E-39
     H     -W*( 4.908923964523425E-41-W*( 1.363041261779140E-42
     I     -W*( 3.682493515461147E-44-W*( 9.687280238870766E-46
     J +FNC ))) ))) ))) )))
      SUM=0.8862269254527580E0-EX*(1.0E0-W*(3.333333333333333E-01
     1     -W*( 9.999999999999999E-02-W*( 2.380952380952381E-02
     2     -W*( 4.629629629629629E-03-W*( 7.575757575757576E-04
     3     -W*( 1.068376068376068E-04-W*( 1.322751322751323E-05
     4     -W*( 1.458916900093371E-06-W*( 1.450385222315047E-07
     5     -W*( 1.312253296380281E-08-W*( 1.089222103714857E-09
     6     -W*( 8.350702795147239E-11-W*( 5.947794013637635E-12
     7     -W*( 3.955429516458526E-13-W*( 2.466827010264457E-14
     8     -W*( 1.448326464359814E-15-W*( 8.032735012415774E-17
     9     -W*( 4.221407208807088E-18-W*( 2.107855191442136E-19
     A     -W*( 1.002516493490772E-20-W*( 4.551846758928200E-22
     B     -W*( 1.977064753877905E-23-W*( 8.230149299214223E-25
     C     -W*( 3.289260349175752E-26-W*( 1.264107898898916E-27
     D +FNB ))))) ))))) ))))) ))))) ))))) )
      COEFT=EXP(W)*SUM
      RETURN
      END
      FUNCTION EFT(EX)
      SRPI=1.772453850905516
      EFT=1.-EXP(-EX**2)*COEFT(EX)*2./SRPI
      RETURN
      END
      SUBROUTINE FLOWP
C*****************************************************************
C PURPOSE: EFFECT OF A FLOWING PLASMA ON ION COLLECTION
C*****************************************************************
C
      RETURN
      END
```

```
      SUBROUTINE PHOTO(N,NANG,ITER,IGEOM,OMEGA,SO,J)
C
C
C
C
C
C
C
C
C                     PACKAGE # 4
C                 ---------------------
C
C CURRENT COLLECTION DUE TO SECONDARY PARTICLES ( PHOTOELECTRONS,
C SECONDARIES,BACKSCATTERS,ELECTRONS INDUCED BY PROTONS)
C
C
C
C
C
C
C
C
C
C
C*******************************************************************
      INTEGER PERSEC,SECIND,PERPAN,PANIND
      COMMON PSAVE(144,10),CSAVE(144,10),PSECTR(144),CSECTR(144),
     1DELEE(20),DELMAX(144),EMAX(144),ETA(20),BSCAT1(144),
     2BSCAT2(144),BSCAT3(144),PHO(20,40),
     1ANG(40),CPHOMX(144),SUNANG,ANGO,BOUNDL(20,40),BOUNDI(20,40),
     2ANGL(20,40),VII(20,40),ICHECK(20,40),RATIO(13),IROOT(144),
     3IFAULT(14),JLOCAL(144),ICON(144),
     1VI(20),EV(8),NLEV,NTIMES,ILEV,TIMIN(20),TIMAX(20),VIMUM(20),
     2A(144),B(144),C(144),BB(144),BM(144),TM(144),DW(144),
     3EW(144),TWOCOS(700),
     1IFLG,NPOINT,M,IGEOM,ISYM,IPOIS,NSECT,NPEROD,MPEROD,IDIMY,
     2GAMMA,DVDX(70,48),DVDY(70,48),Y(144),DELTAX,DELTAY,
     3YS(66,48),XS(66),XSS(70),
     1VARBLE(4),T,DELT,SBOUND,NDIM,POT,H,J1,RPI,RPI2,RPIHAF,SQTPI,
     2SAY,TEMPE1,TEMPE2,DENE1,DENE2,DENI1,DENI2,TEMPI1,TEMPI2,
     3TEMPR,LUN3,LUN4,LUN5,LUN6,LUN7,LUN8,LUN9,LUN10,
     1IDTL(144),JDTL(144),PERSEC(144),SECIND(144),IPANEL(144),
     2PERPAN(144,4),PANIND(144),
     1CION(144),CELEC(144),CSEC(144),CSEC1(144),CPHOTO(144),
     2CPHOT1(144),CBSCAT(144),CBSCA1(144),SECPRN(144),SE'PRI(144),
     3PHI(144),CGUN(144),
     1KASE,JOML,DPHI,NANG,RAD,PBOUND,OMEGA,ALPHA,SO,JP,IS,ECS,
     2EXPVI2(20),EFTVI(20),IERROR,TETMIN,TETMAX,BETA1,BETA2,JLESS,
     3L2,CARRIV,CPH,XI,CSECEL,CSECE1,CS,CB,CS1,CB1
C THIS SUBROUTINE CALCULATES THE PHOTOELECTRON CURRENT CPHOTO
C CPHOTO CONSISTS OF TWO PARTS:CPHOTO,MAIN PHOTOELECTRON CURRENT
C   LEAVING A LOCATION DUE TO DIRECT SUN,AND CARRIV, PHOTOELECTRON
C   CURRENT ARRIVING AT A LOCATION FROM OTHER PORTIONS OF THE CRAFT
C PHI IS POTENTIAL ON SPACECRAFT SURFACE
C N IS NUMBER OF GRID POINTS ON SPACECRAFT SURFACE
C NANG IS NUMBER OF INTERVALS THAT HALF-ANGLE (RPI/2) GRID FOR
C   PHOTOELECTRONS IS DIVIDED INTO. NANG SHOULD BE ODD SO THAT THERE IS
C   A GRID POINT FOR RPI/2
C NLEV IS NUMBER OF VELOCITY LEVELS USED
C CPHOMX IS MAXIMUM CURRENT DUE TO PHOTOELECTRONS
C PHO IS ACTUAL CONTRIBUTION AT EACH POINT DUE TO SUN ANGLE
C ANG IS LOCATION OF EACH GRID POINT. NOTE THAT ANG AND PHO ARE
C   NLEV X NANG MATRICES.
C KL IS A FLAG
C KL=0,BOUNDARIES DETERMINED BY GRID POINTS IN VELOCITY SPACE
C KL=1,BOUNDARIES DETERMINED BY BISSECTION
C*******************************************************************
      DATA KL/0/
      DATA DANG/0.017/
      DATA FAC/1.0/,DVI/0.01/
      GAMMA=RATIO(5)
      DELANG=.5*RPI/FLOAT(NANG-1)
      L1=2*NANG-1
      L2=L1-1
      NDIM=4
      VAR3=Y(J)-SUNANG
      CPHOTO(J)=0.0
      CPHOT1(J)=0.0
      IF(COS(VAR3).NE.ABS(COS(VAR3))) RETURN
C
C
C
C
C GENERATE COEFFICIENTS PHO(I,L) AND ANG(L) FOR USE IN CALCULATING
C   PHOTOELECTRON CURRENT DUE TO PHOTOELECTRONS ARRIVING FROM OTHER
C   PORTIONS OF THE SPACECRAFT
      IOML=1
      DO 300 I=1,NLEV
      DO 190 L=1,L1
      IF(KL.EQ.0) ICHECK(I,L)=1
      IF(KL.EQ.1) ICHECK(I,L)=0
      PHO(I,L)=0.0
      IF(IOML.EQ.0) GO TO 190
      ANG(L)=DELANG*FLOAT(L-1)
      BETA1=VI(I)**2+PHI(J)/GAMMA
      IF(BETA1.LT.0.0) GO TO 180
      VARBLE(1)=SO
      VARBLE(2)=VI(I)*SIN(ANG(L))
      VARBLE(3)=Y(J)
      VARBLE(4)=-VI(I)*COS(ANG(L))
      COSA=COS(ANG(L))
      DBETA=-COSA*DVDX(1,J)/(2.0*VI(I)**2*GAMMA)+COSA
      IF(L.EQ.1.AND.DBETA.LT.0.0) GO TO 180
      IF(L.EQ.L1.AND.DBETA.GT.0.0) GO TO 180
    1 CONTINUE
      T=0.
      CALL ORBIT(SO,BETA1,J)
      IF(VARBLE(1).GE.(SBOUND-DELTAX))GO TO 190
      VAR3=VARBLE(3)-SUNANG
      IF (ABS((POT-PHI(J))/RATIO(5)).LT.100)
     1PHO(I,L)=CPHOMX(J1)*(COS(VAR3)+ABS(COS(VAR3)))
     1*EXP((POT-PHI(J))/RATIO(5))/SQTPI
      IF (PHO(I,L).EQ.0.0) ICHECK(I,L)=1
      GO TO 190
  180 VAR3=Y(J)-SUNANG
      PHO(I,L)= CPHOMX(J)*(COS(VAR3)+ABS(COS(VAR3)))/SQTPI
  190 CONTINUE
      IOML=0
      DO 300 L=1,L1
      IF(PHO(I,L).NE.0.0) IOML=IOML+1
  300 CONTINUE
C
C
C
C
       DO 197 L=1,L1
       DO 197 I=1,NLEV
       BOUNDL(I,L)=PHO(I,L)
       BOUNDI(I,L)=PHO(I,L)
      VII(I,L)=VI(I)
      ANGL(I,L)=ANG(L)
  197 CONTINUE
C
      IF(KL.EQ.0) GO TO 2100
      DO 200 I=1,NLEV
      DO 200 L=1,L2
      IF (ICHECK(I,L).EQ.1.OR.ICHECK(I,L+1).EQ.1) GO TO 200
      IF (PHO(I,L).EQ.0.0.AND.PHO(I,L+1).NE.0.0) GO TO 201
      IF (PHO(I,L).NE.0.0.AND.PHO(I,L+1).EQ.0.0) GO TO 202
      GO TO 200
  201 LCUT=L
      ANGA=ANG(L)
      ANGB=ANG(L+1)
      GO TO 203
  202 LCUT=L+1
      ANGA=ANG(L+1)
      ANGB=ANG(L)
  203 ANGLE=0.5*(ANGA+ANGB)
  206 AOLD=ANGLE
      VARBLE(1)=SO
      VARBLE(2)=ANGLE
      VARBLE(3)=Y(J)
      VAR3=VARBLE(3) SUNANG
      BETA1=VI(I)*VI(I)+PHI(J)/GAMMA
      IF(BETA1.LT.0.0) BOUNDL(I,LCUT)=CPHOMX(J)*(COS(VAR3)+
     1 ABS(COS(VAR3)))/SQTPI
      IF(BETA1.LT.0.0) GO TO 207
      T=0.0
      CALL ORBIT(SO,BETA1,J)
      IF (VARBLE(1).GE.(SBOUND-DELTAX)) GO TO 204
      VAR3=VARBLE(3)-SUNANG
      ISURF=1
      IF(ABS((PHI(J)-POT)/GAMMA).LT.100.0)
     1 BOUNDL(I,LCUT)=CPHOMX(J1)*(COS(VAR3)+ABS(COS(VAR3)))
     2 *EXP((POT-PHI(J))/GAMMA)/SQTPI
      ANGB=ANGLE
      GO TO 205
  204 ANGA=ANGLE
  205 ANGLE=0.5*(ANGA+ANGB)
      IF (ABS(ANGLE-AOLD).GT.DANG.OR.ISURF.EQ.0) GO TO 206
  207 ANGL(I,LCUT)=ANGLE
  200 CONTINUE
C
C THEN DO A REFINEMENT AS FUNCTION OF VELOCITY LEVELS.
      ISURF=0
      NLEV1=NLEV-1
      DO 210 L=1,L1
      DO 210 I=1,NLEV1
      IF (ICHECK(I,L).EQ.1.OR.ICHECK(I+1,L).EQ.1) GO TO 210
      IF (PHO(I,L).EQ.0.0.AND.PHO(I+1,L).NE.0.0) GO TO 211
      IF (PHO(I,L).NE.0.0.AND.PHO(I+1,L).EQ.0.0) GO TO 212
       GO TO 210
  211 ICUT=I
      VIA=VI(I)
      VIB=VI(I+1)
      GO TO 213
  212 ICUT=I+1
      VIA=VI(I+1)
      VIB=VI(I)
  213 VIM=0.5*(VIA+VIB)
  216 VIOLD=VIM
      BETA1= VIM**2+PHI(J)/GAMMA
      VARBLE(1)=SO
      VARBLE(2)=ANG(L)
      VARBLE(3)=Y(J)
      IF(BETA1.LE.0.0) BOUNDI(ICUT,L)=CPHOMX(J)*(COS(VAR3)+
     1 ABS(COS(VAR3)))/SQTPI
      IF(BETA1.LE.0) GO TO 217
      T=0.0
      CALL ORBIT(SO,BETA1,J)
      IF (VARBLE(1).GE.(SBOUND-DELTAX)) GO TO214
      VAR3=VARBLE(3)-SUNANG
      ISURF=1
      IF(ABS((PHI(J)-POT)/GAMMA).LT.100.0)
     1 BOUNDI(ICUT,L)=CPHOMX(J1)*(COS(VAR3)+ABS(COS(VAR3)))
     1 *EXP((POT-PHI(J))/GAMMA)/SQTPI
      VIB=VIM
      GO TO 215
  214 VIA=VIM
      GO TO 215
  215 VIM=0.5*(VIA+VIB)
      IF(ABS(VIM-VIOLD).GT.DVI*(VI(I+1)-VI(1)).OR.ISURF.EQ.0) GO TO 216
  217 VII(ICUT,L)=VIM
  210 CONTINUE
C
C
C
 2100 CONTINUE
C
C CALCULATE CPHOTO
      CURMIN=0.
      CURMAX=0.
      CARRIV=0.
      VAR4=Y(J)-SUNANG
      CPHOTO(J)=.5*CPHOMX(J)*(COS(VAR4)+ABS(COS(VAR4)))
      CPHOT1(J)=CPHOTO(J)
      IF(IFAULT(6).EQ.0)GO TO 4000
      CPH=0.0
      DO 350 L=1,L1
      IF(PHO(1,L).NE.0.0.OR.PHI(J).GT.0.0) CPH=2.0*CPHOTO(J)/SQTPI
  350 CONTINUE
      WRITE (LUN6,6000) J
 6000 FORMAT (/1X,' SAMPLING IN VELOVITY SPACE FOR PHOTO ELEC AT J=',I3/
     1 )
      WRITE (LUN6,600) ((PHO(I,L),I=1,NLEV),L=1,L1)
  600 FORMAT (8(1X,1PE9.2))
      WRITE (LUN6,601)
      IF(J.NE.1) GO TO 359
      WRITE (LUN6,600) ((BOUNDL(I,L),I=1,NLEV),L=1,L1)
      WRITE (LUN6,601)
      WRITE (LUN6,600)((BOUNDI(I,L),I=1,NLEV),L=1,L1)
      WRITE (LUN6,601)
      WRITE (LUN6,600) ((VII(I,L),I=1,NLEV),L=1,L1)
      WRITE (LUN6,601)
      WRITE (LUN6,600) ((ANGL(I,L),I=1,NLEV),L=1,L1)
      WRITE (LUN6,601)
      WRITE (LUN6,600) (VI(I),I=1,NLEV)
      WRITE (LUN6,602) ((ICHECK(I,L),I=1,NLEV),L=1,L1)
  602 FORMAT (8I8)
  359 CONTINUE
      CALL FIT(L2,CARRIV,CURMIN,CURMAX,J,CPH,PHI,N,GAMMA)
 4000 IF(PHI(J).GT.0.0)
     1 CPHOTO(J)=CPHOTO(J)*2.0*SAY*(SQRT(-PHI(J)/RATIO(5))+
     2 COEFT(SQRT(-PHI(J)/RATIO(5))))
     3 *EXP(PHI(J)/RATIO(5))
  601 FORMAT(//)
  360 CPHOTO(J)=CPHOTO(J)-CARRIV
  361 FORMAT(2(1X,1PE10.3))
  400 CONTINUE
      RETURN
      END
      SUBROUTINE SECON(N,NANG,PHI,CSEC,ITER,IGEOM,OMEGA,SO,J,
     1 CSEC1)
```

```
      INTEGER PERSEC,SECIND,PERPAN,PANIND
      COMMON PSAVE(144,10),CSAVE(144,10),PSECTR(144),CSECTR(144),
     1DELEE(20),DELMAX(144),EMAX(144),ETA(20),BSCAT1(144),
     2BSCAT2(144),BSCAT3(144),PHO(20,40),
     1ANG(40),CPHOMX(144),SUNANG,ANGO,BOUNDL(20,40),BOUNDI(20,40),
     2ANGL(20,40),VII(20,40),ICHECK(20,40),RATIO(13),IROOT(144),
     3IFAULT(14),JLOCAL(144),ICON(144),
     1VI(20),EV(8),NLEV,NTIMES,ILEV,TIMIN(20),TIMAX(20),VINUM(20),
     2A(144),B(144),C(144),BB(144),BW(144),TW(144),DW(144),
     3EW(144),TWOCOS(700),
     1IFLG,NPOINT,N,IGEOM,ISYM,IPOIS,NSECT,NPEROD,MPEROD,IDIMY,
     2GAMMA,DVDX(70,48),DVDY(70,48),Y(144),DELTAX,DELTAY,
     3YS(66,48),XS(66),XSS(70),
     1VARBLE(4),T,DELT,SBOUND,NDIM,POT,H,J1,RPI,RPI2,RPIHAF,SQTPI,
     2SAY,TEMPE1,TEMPE2,DENE1,DENE2,DENI1,DENI2,TEMPI1,TEMPI2,
     3TEMPR,LUN3,LUN4,LUN5,LUN6,LUN7,LUN8,LUN9,LUN10,
     1IDTL(144),JDTL(144),PERSEC(144),SECIND(144),IPANEL(144),
     2PERPAN(144,4),PANIND(144),
     1CION(144),CELEC(144),CSEC(144),CSEC1(144),CPHOTO(144),
     2CPHOT1(144),CBSCAT(144),CBSCA1(144),SECPRN(144),SECPR1(144),
     3PHI(144),CGUN(144),
     1KASE,JOML,DPHI,NANG,RAD,PBOUND,OMEGA,ALPHA,SO,JP,IS,ECS,
     2EXPVI2(20),EFTVI(20),IERROR,TETMIN,TETMAX,BETA1,BETA2,JLESS,
     3L2,CARRIV,CPH,XI,CSECEL,CSECE1,CS,CB,CS1,CB1
C THIS SUBROUTINE CALCULATES THE SECONDARY CURRENT CSEC
C CSEC CONSISTS OF TWO PARTS:CSEL,MAIN SECONDARY CURRENT
C   LEAVING A LOCATION,AND CARRIV,SECONDARY
C   CURRENT ARRIVING AT A LOCATION FROM OTHER PORTIONS OF THE CRAFT
C PHI IS POTENTIAL ON SPACECRAFT SURFACE
C N IS NUMBER OF GRID POINTS ON SPACECRAFT SURFACE
C NANG IS NUMBER OF INTERVALS THAT HALF-ANGLE (RPI/2) GRID FOR
C   SECONDARYS IS DIVIDED INTO. NANG SHOULD BE ODD SO THAT THERE IS
C   A GRID POINT FOR RPI/2
C NLEV IS NUMBER OF VELOCITY LEVELS USED
C PHO IS ACTUAL CONTRIBUTION AT EACH POINT
C ANG IS LOCATION OF EACH GRID POINT. NOTE THAT ANG AND PHO ARE
C  NLEV X NANG MATRICES.
      DATA DANG/0.017/
      DATA FAC/1.0/,DVI/0.01/
      DATA KL/0/
      GAMMA=RATIO(3)
      NDIM=3
      DELANG=.5*RPI/FLOAT(NANG-1)
      L1=2*NANG-1
      L2=L1-1
C GENERATE COEFFICIENTS PHO(I,L) AND ANG(L) FOR USE IN CALCULATING
C   SECONDARY CURRENT DUE TO SECONDARYS ARRIVING FROM OTHER
C   PORTIONS OF THE SPACECRAFT
      IOML=1
      DO 300 I=1,NLEV
      DO 191 L=1,L1
      IF(KL.EQ.0) ICHECK(I,L)=1
      IF(KL.EQ.1) ICHECK(I,L)=0
      PHO(I,L)=0.0
      IF(IOML.EQ.0) GO TO 191
      ANG(L)=DELANG*FLOAT(L-1)
      BETA1=VI(I)**2+PHI(J)/GAMMA
      IF(BETA1.LT.0.0) GO TO 180
      VARBLE(1)=SO
      VARBLE(2)=ANG(L)
      VARBLE(3)=Y(J)
      COSA=COS(ANG(L))
      DBETA=-COSA*DVDX(1,J)/(2.0*VI(I)**2*GAMMA)+COSA
      IF(L.EQ.1.AND.DBETA.LT.0.) GOTO 180
      IF (L.EQ.L1.AND.DBETA.GT.0.) GO TO 180
      T=0.
      CALL ORBIT(SO,BETA1,J)
      IF(VARBLE(1).GE.(SBOUND-DELTAX))GO TO 190
      IF (ABS((PHI(J)-POT)/RATIO(3)).LT.100.)
     1 PHO(I,L)=2.0*CSEC1(J1)*EXP((POT-PHI(J))/RATIO(3))/
     2 SQTPI
      IF (PHO(I,L).EQ.0.0) ICHECK(I,L)=1
      GO TO 190
  180 PHO(I,L)=2.0*CSEC1(J)/SQTPI
  190 CONTINUE
  191 CONTINUE
      IOML=0
      DO 300 L=1,L1
      IF(PHO(I,L).NE.0.0) IOML=IOML+1
  300 CONTINUE
C
C
       DO 197 L=1,L1
       DO 197 I=1,NLEV
       BOUNDL(I,L)=PHO(I,L)
       BOUNDI(I,L)=PHO(I,L)
      VII(I,L)=VI(I)
      ANGL(I,L)=ANG(L)
  197 CONTINUE
C
C
C
      IF(KL.EQ.0) GO TO 2100
      DO 200 I=1,NLEV
      DO 200 L=1,L2
      IF (ICHECK(I,L).EQ.1.OR.ICHECK(I,L+1).EQ.1) GO TO 200
      IF (PHO(I,L).EQ.0.0.AND.PHO(I,L+1).NE.0.0) GO TO 201
      IF (PHO(I,L).NE.0.0.AND.PHO(I,L+1).EQ.0.0) GO TO 202
      GO TO 200
  201 LCUT=L
      ANGA=ANG(L)
      ANGB=ANG(L+1)
      GO TO 203
  202 LCUT=L+1
      ANGA=ANG(L+1)
      ANGB=ANG(L)
  203 ANGLE=0.5*(ANGA+ANGB)
  206 AOLD=ANGLE
      VARBLE(1)=SO
      VARBLE(2)=ANGLE
      VARBLE(3)=Y(J)
      T=0.0
      BETA1=VI(I)*VI(I)+PHI(J)/GAMMA
      IF(BETA1.LT.0.0) BOUNDL(I,LCUT)=2.0*CSEC1(J)/SQTPI
      IF(BETA1.LT.0.0) GO TO 207
      CALL ORBIT(SO,BETA1,J)
      IF (VARBLE(1).GE.(SBOUND-DELTAX)) GO TO 204
      IF(ABS((PHI(J)-POT)/GAMMA).LT.100.)
     1BOUNDL(I,LCUT)=2.0*CSEC1(J1)*EXP((POT-PHI(J))/GAMMA)/SQTPI
  207 ANGB=ANGLE
      GO TO 205
  204 ANGA=ANGLE
  205 ANGLE=0.5*(ANGA+ANGB)
      IF (ABS(ANGLE-AOLD).GT.DANG) GO TO 206
      ANGL(I,LCUT)=ANGLE
  200 CONTINUE
C
C
      NLEV1=NLEV-1
      DO 210 L=1,L1
      DO 210 I=1,NLEV1
      IF (ICHECK(I,L).EQ.1.OR.ICHECK(I+1,L).EQ.1) GO TO 210
      IF (PHO(I,L).EQ.0.0.AND.PHO(I+1,L).NE.0.0) GO TO 211
      IF (PHO(I,L).NE.0.0.AND.PHO(I+1,L).EQ.0.0) GO TO 212
      GO TO 210
  211 ICUT=I
      VIA=VI(I)
      VIB=VI(I+1)
      GO TO 213
  212 ICUT=I+1
      VIA=VI(I+1)
      VIB=VI(I)
  213 VIM=0.5*(VIA+VIB)
  216 VIOLD=VIM
      BETA1= VIM**2+PHI(J)/GAMMA
      IF(BETA1.LT.0.0) BOUNDI(ICUT,L)=2.0*CSEC1(J)/SQTPI
      IF (BETA1.LE.0.0) GO TO 217
      VARBLE(1)=SO
      VARBLE(2)=ANG(L)
      VARBLE(3)=Y(J)
      T=0.0
      CALL ORBIT(SO,BETA1,J)
      IF (VARBLE(1).GE.(SBOUND-DELTAX)) GO TO214
      IF(ABS((PHI(J)-POT)/GAMMA).LT.100.)
     1BOUNDI(ICUT,L)=2.0*CSEC1(J1)*EXP((POT-PHI(J))/GAMMA)/SQTPI
  217 VIB=VIM
      GO TO 215
  214 VIA=VIM
      GO TO 215
  215 VIM=0.5*(VIA+VIB)
      IF(ABS(VIM-VIOLD).GT.DVI*(VI(I+1)-VI(I))) GO TO 216
      VII(ICUT,L)=VIM
  210 CONTINUE
C
C
C
 2100 CONTINUE
C
C CALCULATE CSEC
      CURMIN=0.
      CURMAX=0.
      CARRIV=0.
      IF(IFAULT(7).EQ.0)GO TO 4000
      CPH=0.0
      DO 350 L=1,L1
      IF(PHO(1,L).NE.0.0.OR.PHI(J).GT.0.0)
     1 CPH=2.0*CSEC1(J)/SQTPI
  350 CONTINUE
      WRITE (LUN6,6000) J
 6000 FORMAT (/1X,*SAMPLING IN VELOCITY SPACE FOR SECONDARIES FOR J=*,
     1 I3/)
      WRITE (LUN6,600) ((PHO(I,L),I=1,NLEV),L=1,L1)
  600 FORMAT (8(1X,1PE9.2))
  359 CONTINUE
      CALL FIT(L2,CARRIV,CURMIN,CURMAX,J,CPH,PHI,N,GAMMA)
 4000 CONTINUE
  360 CSEC(J)=CSEC(J)-CARRIV
  400 CONTINUE
      RETURN
      END
      SUBROUTINE BSCATR(N,NANG,PHI,CBSCAT,ITER,IGEOM,OMEGA,SO,J,CBSCA1)
      INTEGER PERSEC,SECIND,PERPAN,PANIND
      COMMON PSAVE(144,10),CSAVE(144,10),PSECTR(144),CSECTR(144),
     1DELEE(20),DELMAX(144),EMAX(144),ETA(20),BSCAT1(144),
     2BSCAT2(144),BSCAT3(144),PHO(20,40),
     1ANG(40),CPHOMX(144),SUNANG,ANGO,BOUNDL(20,40),BOUNDI(20,40),
     2ANGL(20,40),VII(20,40),ICHECK(20,40),RATIO(13),IROOT(144),
     3IFAULT(14),JLOCAL(144),ICON(144),
     1VI(20),EV(8),NLEV,NTIMES,ILEV,TIMIN(20),TIMAX(20),VINUM(20),
     2A(144),B(144),C(144),BB(144),BW(144),TW(144),DW(144),
     3EW(144),TWOCOS(700),
     1IFLG,NPOINT,N,IGEOM,ISYM,IPOIS,NSECT,NPEROD,MPEROD,IDIMY,
     2GAMMA,DVDX(70,48),DVDY(70,48),Y(144),DELTAX,DELTAY,
     3YS(66,48),XS(66),XSS(70),
     1VARBLE(4),T,DELT,SBOUND,NDIM,POT,H,J1,RPI,RPI2,RPIHAF,SQTPI,
     2SAY,TEMPE1,TEMPE2,DENE1,DENE2,DENI1,DENI2,TEMPI1,TEMPI2,
     3TEMPR,LUN3,LUN4,LUN5,LUN6,LUN7,LUN8,LUN9,LUN10,
     1IDTL(144),JDTL(144),PERSEC(144),SECIND(144),IPANEL(144),
     2PERPAN(144,4),PANIND(144),
     1CION(144),CELEC(144),CSEC(144),CSEC1(144),CPHOTO(144),
     2CPHOT1(144),CBSCAT(144),CBSCA1(144),SECPRN(144),SECPR1(144),
     3PHI(144),CGUN(144),
     1KASE,JOML,DPHI,NANG,RAD,PBOUND,OMEGA,ALPHA,SO,JP,IS,ECS,
     2EXPVI2(20),EFTVI(20),IERROR,TETMIN,TETMAX,BETA1,BETA2,JLESS,
     3L2,CARRIV,CPH,XI,CSECEL,CSECE1,CS,CB,CS1,CB1
C THIS SUBROUTINE CALCULATES THE BSCATTER CURRENT CBSCAT
C CBSCAT CONSISTS OF TWO PARTS:CBSCAT,MAIN BSCATTER CURRENT
C   LEAVING A LOCATION,AND CARRIV,BSCATTER
C   CURRENT ARRIVING AT A LOCATION FROM OTHER PORTIONS OF THE CRAFT
C PHI IS POTENTIAL ON SPACECRAFT SURFACE
C N IS NUMBER OF GRID POINTS ON SPACECRAFT SURFACE
C NANG IS NUMBER OF INTERVALS THAT HALF-ANGLE (RPI/2) GRID FOR
C   BSCATTERS IS DIVIDED INTO. NANG SHOULD BE ODD SO THAT THERE IS
C   A GRID POINT FOR RPI/2
C NLEV IS NUMBER OF VELOCITY LEVELS USED
C PHO IS ACTUAL CONTRIBUTION AT EACH POINT
C ANG IS LOCATION OF EACH GRID POINT. NOTE THAT ANG AND PHO ARE
C  NLEV X NANG MATRICES.
      DATA DANG/0.017/
      DATA FAC/1.0/,DVI/0.01/
      DATA KL/0/
      GAMMA=RATIO(4)
      DELANG=.5*RPI/FLOAT(NANG-1)
      L1=2*NANG-1
      L2=L1-1
      NDIM=3
C GENERATE COEFFICIENTS PHO(I,L) AND ANG(L) FOR USE IN CALCULATING
C   BSCATTER CURRENT DUE TO BSCATTERS ARRIVING FROM OTHER
C   PORTIONS OF THE SPACECRAFT
      IOML=1
      DO 300 I=1,NLEV
      DO 190 L=1,L1
      IF(KL.EQ.0) ICHECK(I,L)=1
      IF(KL.EQ.1) ICHECK(I,L)=0
      PHO(I,L)=0.0
      IF(IOML.EQ.0) GO TO 190
      ANG(L)=DELANG*FLOAT(L-1)
      BETA1=VI(I)**2+PHI(J)/GAMMA
      IF(BETA1.LT.0.0) GO TO 180
      VARBLE(1)=SO
      VARBLE(2)=ANG(L)
      VARBLE(3)=Y(J)
      COSA=COS(ANG(L))
      DBETA=-COSA*DVDX(1,J)/(2.0*VI(I)**2*GAMMA)+COSA
      IF (L.EQ.1.AND.DBETA.LT.0.0) GO TO 180
      IF (L.EQ.L1.AND.DBETA.GT.0.0) GO TO 180
      T=0.
      CALL ORBIT(SO,BETA1,J)
      IF(VARBLE(1).GE.(SBOUND-DELTAX))GO TO 190
      IF (ABS((PHI(J)-POT)/RATIO(4)).LT.200.)
     1 PHO(I,L)=2.0*CBSCA1(J1)*EXP((POT-PHI(J))/RATIO(4))/
     5 SQTPI
      IF (PHO(I,L).EQ.0.0) ICHECK(I,L)=1
      GO TO 190
```

```
  180 PHO(I,L)=2.0*CBSCA1(J)/SQTPI
  190 CONTINUE
      IOML=0
      DO 300 L=1,L1
      IF(PHO(I,L).NE.0.0) IOML=IOML+1
  300 CONTINUE
C
C
       DO 197 L=1,L1
       DO 197 I=1,NLEV
       BOUNDL(I,L)=PHO(I,L)
       BOUNDI(I,L)=PHO(I,L)
      VII(I,L)=VI(I)
      ANGL(I,L)=ANG(L)
  197 CONTINUE
C
C
C
C
      IF(KL.EQ.0) GO TO 2100
      DO 200 I=1,NLEV
      DO 200 L=1,L2
      IF (ICHECK(I,L).EQ.1.OR.ICHECK(I,L+1).EQ.1) GO TO 200
      IF (PHO(I,L).EQ.0.0.AND.PHO(I,L+1).NE.0.0) GO TO 201
      IF (PHO(I,L).NE.0.0.AND.PHO(I,L+1).EQ.0.0) GO TO 202
      GO TO 200
  201 LCUT=L
      ANGA=ANG(L)
      ANGB=ANG(L+1)
      GO TO 203
  202 LCUT=L+1
      ANGA=ANG(L+1)
      ANGB=ANG(L)
  203 ANGLE=0.5*(ANGA+ANGB)
  206 AOLD=ANGLE
      VARBLE(1)=SO
      VARBLE(2)=ANGLE
      VARBLE(3)=Y(J)
      T=0.0
      BETA1=VI(I)*VI(I)+PHI(J)/GAMMA
      IF(BETA1.LT.0.0) BOUNDL(I,LCUT)=2.0*CBSCA1(J)/SQTPI
      IF(BETA1.LT.0.) GO TO 207
      CALL ORBIT(SO,BETA1,J)
      IF (VARBLE(1).GE.(SBOUND-DELTAX)) GO TO 204
      IF(ABS((PHI(J)-POT)/GAMMA).LT.100.0)
     1BOUNDL(I,LCUT)=2.0*CBSCA1(J1)*EXP((POT-PHI(J))/GAMMA)/SQTPI
  207 ANGB=ANGLE
      GO TO 205
  204 ANGA=ANGLE
  205 ANGLE=0.5*(ANGA+ANGB)
      IF (ABS(ANGLE-AOLD).GT.DANG) GO TO 206
      ANGL(I,LCUT)=ANGLE
  200 CONTINUE
C
      NLEV1=NLEV-1
      DO 210 L=1,L1
      DO 210 I=1,NLEV1
      IF (ICHECK(I,L).EQ.1.OR.ICHECK(I+1,L).EQ.1) GO TO 210
      IF (PHO(I,L).EQ.0.0.AND.PHO(I+1,L).NE.0.0) GO TO 211
      IF (PHO(I,L).NE.0.0.AND.PHO(I+1,L).EQ.0.0) GO TO 212
       GO TO 210
  211 ICUT=I
      VIA=VI(I)
      VIB=VI(I+1)
      GO TO 213
  212 ICUT=I+1
      VIA=VI(I+1)
      VIB=VI(I)
  213 VIM=0.5*(VIA+VIB)
  216 VIOLD=VIM
      BETA1= VIM**2+PHI(J)/GAMMA
      IF(BETA1.LT.0.0) BOUNDI(ICUT,L)=2.0*CBSCA1(J)/SQTPI
      IF (BETA1.LT.0.0) GO TO 217
      VARBLE(1)=SO
      VARBLE(2)=ANG(L)
      VARBLE(3)=Y(J)
      T=0.0
      CALL ORBIT(SO,BETA1,J)
      IF (VARBLE(1).GE.(SBOUND-DELTAX)) GO TO214
      IF(ABS((PHI(J)-POT)/GAMMA).LT.100.0)
     1BOUNDI(ICUT,L)=2.0*CBSCA1(J1)*EXP((POT-PHI(J))/GAMMA)/SQTPI
  217 VIB=VIM
      GO TO 215
  214 VIA=VIM
      GO TO 215
  215 VIM=0.5*(VIA+VIB)
      IF(ABS(VIM-VIOLD).GT.DVI*(VI(I+1)-VI(I))) GO TO 216
      VII(ICUT,L)=VIM
  210 CONTINUE
C
C
C
 2100 CONTINUE
C CALCULATE CBSCAT
      CURMIN=0.
      CURMAX=0.
      CARRIV=0.
      IF(IFAULT(8).EQ.0)GO TO 4000
      CPH=0.0
      DO 350 L=1,L1
      IF(PHO(I,L).NE.0.0.OR.PHI(J).GT.0.0)
     1 CPH=2.0*CBSCA1(J)/SQTPI
  350 CONTINUE
      WRITE (LUN6,6000) J
 6000 FORMAT (/1X,*SAMPLING IN VELOCITY SPACE FOR BSCATTERS AT J=*,
     1 I3/)
      WRITE (LUN6,600) ((PHO(I,L),I=1,NLEV),L=1,L1)
  600 FORMAT (8(1X,1PE9.2))
      CALL FIT(L2,CARRIV,CURMIN,CURMAX,J,CPH,PHI,N,GAMMA)
 4000 CONTINUE
  601 FORMAT(//)
  360 CBSCAT(J)=CBSCAT(J)-CARRIV
  400 CONTINUE
      RETURN
      END
      SUBROUTINE ELPRN(N,NANG,PHI,SECPRN,ITER,IGEOM,OMEGA,SO,J,SECPR1)
      INTEGER PERSEC,SECIND,PERPAN,PANIND
      COMMON PSAVE(144,10),CSAVE(144,10),PSECTR(144),CSECTR(144),
     1DFLEE(20),DELMAX(144),EMAX(144),ETA(20),BSCAT1(144),
     2BSCAT2(144),BSCAT3(144),PHO(20,40),
     1ANG(40),CPHOMX(144),SUMANG,ANGO,BOUNDL(20,40),BOUNDI(20,40),
     2ANGL(20,40),VII(20,40),ICHECK(20,40),RATIO(13),IROOT(144),
     3IFAULT(14),JLOCAL(144),ICON(144),
     1VI(20),EV(8),NLEV,NTIMES,ILEV,TIMIN(20),TIMAX(20),VIMUM(20),
     2A(144),B(144),C(144),BB(144),BW(144),TW(144),DW(144),
     3FW(144),TWOCOS(700),
     1IFLG,NPOINT,M,IGEOM,ISYM,IPOIS,NSECT,NPEROD,MPEROD,IDIMY,
     2GAMMA,DVDX(70,48),DVDY(70,48),Y(144),DELTAX,DELTAY,
     3YS(66,48),XS(66),XSS(70),
     1VARBLE(4),T,DELT,SBOUND,NDIM,POT,N,J1,RPI,RPI2,RPIHAF,SQTPI,
     2SAY,TEMPE1,TEMPE2,DENE1,DENE2,DENI1,DENI2,TEMPI1,TEMPI2,
     1IDTL(144),JDTL(144),PERSEC(144),SECIND(144),IPANEL(144),
     2PERPAN(144,4),PANIND(144),
     1CION(144),CELEC(144),CSEC(144),CSEC1(144),CPHOTO(144),
     2CPHOT1(144),CBSCAT(144),CBSCA1(144),SECPRN(144),SECPR1(144),
     3PHI(144),CGUN(144),
     1KASE,JOML,DPHI,NANG,RAD,FBOUND,OMEGA,ALPHA,SO,JP,IS,ECS,
     2EXPVI2(20),EFTVI(20),IERROR,TETMIN,TETMAX,BETA1,BETA2,JLESS,
     3L2,CARRIV,CPH,XI,CSECEL,CSECE1,CS,CB,CS1,CB1
C*********************************************************************
C THIS SUBROUTINE CALCULATES THE SECONDARY CURRENT SECPRN
C SECPRN CONSISTS OF TWO PARTS:SECPRN,MAIN SECONDARY CURRENT
C    LEAVING A LOCATION,AND CARRIV,SECONDARY
C    CURRENT ARRIVING AT A LOCATION FROM OTHER PORTIONS OF THE CRAFT
C PHI IS POTENTIAL ON SPACECRAFT SURFACE
C N IS NUMBER OF GRID POINTS ON SPACECRAFT SURFACE
C NANG IS NUMBER OF INTERVALS THAT HALF-ANGLE (RPI/2) GRID FOR
C    SECONDARYS IS DIVIDED INTO. NANG SHOULD BE ODD SO THAT THERE IS
C    A GRID POINT FOR RPI/2
C NLEV IS NUMBER OF VELOCITY LEVELS USED
C PHO IS ACTUAL CONTRIBUTION AT EACH POINT
C ANG IS LOCATION OF EACH GRID POINT. NOTE THAT ANG AND PHO ARE
C  NLEV X NANG MATRICES.
C**********************************************************************
      DATA DANG/0.1/
      DATA FAC/1.0/,DVI/0.017/
      DATA KL/0/
      GAMMA=RATIO(13)
      DELANG=.5*RPI/FLOAT(NANG-1)
      L1=2*NANG-1
      L2=L1-1
      NDIM=3
C GENERATE COEFFICIENTS PHO(I,L) AND ANG(L) FOR USE IN CALCULATING
C    SECONDARY CURRENT DUE TO SECONDARYS ARRIVING FROM OTHER
C    PORTIONS OF THE SPACECRAFT
      IOML=1
      DO 300 I=1,NLEV
      DO 190 L=1,L1
      IF(KL.EQ.0) ICHECK(I,L)=1
      IF(KL.EQ.1) ICHECK(I,L)=0
      PHO(I,L)=0.0
      IF(IOML.EQ.0) GO TO 190
      ANG(L)=DELANG*FLOAT(L-1)
      BETA1=VI(I)**2+PHI(J)/GAMMA
      IF(BETA1.LT.0.0) GO TO 180
      VARBLE(1)=SO
      VARBLE(2)=ANG(L)
      VARBLE(3)=Y(J)
      COSA=COS(ANG(L))
      DBETA=-COSA*DVDX(1,J)/(2.0*VI(I)**2*GAMMA)+COSA
      IF(L.EQ.1.AND.DBETA.LT.0.0) GO TO 180
      IF(L.EQ.L1.AND.DBETA.GT.0.0) GO TO 180
      T=0.
      CALL ORBIT(SO,BETA1,J)
      IF(VARBLE(1).GE.(SBOUND-DELTAX))GO TO 190
      IF (ABS((POT-PHI(J))/RATIO(13)).LT.100.)
     1 PHO(I,L)=2.0*SECPR1(J1)*EXP((POT-PHI(J))/RATIO(13))/
     5 SQTPI
      IF (PHO(I,L).EQ.0.0) ICHECK(I,L)=1
      GO TO 190
  180 PHO(I,L)=2.0*SECPR1(J)/SQTPI
  190 CONTINUE
      IOML=0
      DO 300 L=1,L1
      IF(PHO(I,L).NE.0.0) IOML=IOML+1
  300 CONTINUE
C
C
       DO 197 L=1,L1
       DO 197 I=1,NLEV
       BOUNDL(I,L)=PHO(I,L)
       BOUNDI(I,L)=PHO(I,L)
      VII(I,L)=VI(I)
      ANGL(I,L)=ANG(L)
  197 CONTINUE
C
C
C
C
      IF(KL.EQ.0) GO TO 2100
      DO 200 I=1,NLEV
      DO 200 L=1,L2
      IF (ICHECK(I,L).EQ.1.OR.ICHECK(I,L+1).EQ.1) GO TO 200
      IF (PHO(I,L).EQ.0.0.AND.PHO(I,L+1).NE.0.0) GO TO 201
      IF (PHO(I,L).NE.0.0.AND.PHO(I,L+1).EQ.0.0) GO TO 202
      GO TO 200
  201 LCUT=L
      ANGA=ANG(L)
      ANGB=ANG(L+1)
      GO TO 203
  202 LCUT=L+1
      ANGA=ANG(L+1)
      ANGB=ANG(L)
  203 ANGLE=0.5*(ANGA+ANGB)
  206 AOLD=ANGLE
      VARBLE(1)=SO
      VARBLE(2)=ANGLE
      VARBLE(3)=Y(J)
      T=0.0
      BETA1=VI(I)*VI(I)+PHI(J)/GAMMA
      IF(BETA1.LT.0.0) BOUNDL(I,LCUT)=2.0*SECPR1(J)/SQTPI
      IF(BETA1.LT.0.0) GO TO 207
      CALL ORBIT(SO,BETA1,J)
      IF (VARBLE(1).GE.(SBOUND-DELTAX)) GO TO 204
      ANGB=ANGLE
      BOUNDL(I,LCUT)=2.0*SECPR1(J1)*EXP((POT-PHI(J))/GAMMA)/SQTPI
      GO TO 205
  204 ANGA=ANGLE
  205 ANGLE=0.5*(ANGA+ANGB)
      IF (ABS(ANGLE-AOLD).GT.DANG) GO TO 206
  207 ANGL(I,LCUT)=ANGLE
  200 CONTINUE
C
      NLEV1=NLEV-1
      DO 210 L=1,L1
      DO 210 I=1,NLEV1
      IF (ICHECK(I,L).EQ.1.OR.ICHECK(I+1,L).EQ.1) GO TO 210
      IF (PHO(I,L).EQ.0.0.AND.PHO(I+1,L).NE.0.0) GO TO 211
      IF (PHO(I,L).NE.0.0.AND.PHO(I+1,L).EQ.0.0) GO TO 212
       GO TO 210
  211 ICUT=I
      VIA=VI(I)
      VIB=VI(I+1)
      GO TO 213
  212 ICUT=I+1
      VIA=VI(I+1)
      VIB=VI(I)
  213 VIM=0.5*(VIA+VIB)
  216 VIOLD=VIM
      BETA1= VIM**2+PHI(J)/GAMMA
      IF(BETA1.LT.0.0) BOUNDI(ICUT,L)=2.0*SECPR1(J)/SQTPI
      IF(BETA1.LT.0.0) GO TO 217
```

```
      VARBLE(2)=ANG(L)
      VARBLE(3)=Y(J)
      T=0.0
      CALL ORBIT(SO,BETA1,J)
      IF (VARBLE(1).GE.(SBOUND-DELTAX)) GO TO214
      VIB=VIM
      SECP=(1.0-WEIGHT)*SECPR1(J1)+WEIGHT*SECPR1(J2)
      BOUNDI(ICUT,L)=2.0*SECP*EXP((POT-PHI(J))/GAMMA)/SQTPI
      GO TO 215
  214 VIA=VIM
      GO TO 215
  215 VIM=0.5*(VIA+VIB)
      IF(ABS(VIM-VIOLD).GT.DVI*(VI(I+1)-VI(I))) GO TO 216
  217 VII(ICUT,L)=VIM
  210 CONTINUE
C
C
C
 2100 CONTINUE
C
C CALCULATE SECPRN
      CURMIN=0.
      CURMAX=0.
      CARRIV=0.
      IF(IFAULT(8).EQ.0)GO TO 4000
      CPH=0.0
      DO 350 L=1,L1
      IF(PHO(1,L).NE.0.0.OR.PHI(J).GT.0.0)
     1 CPH=2.0*SECPR1(J)/SQTPI
  350 CONTINUE
      WRITE (LUN6,6000) J
 6000 FORMAT (1X,*SAMPLING IN VELOCITY SPACE FOR ELEC INDUCED AT J=*,
     1 I3/)
      WRITE (LUN6,600) ((PHO(I,L),I=1,NLEV),L=1,L1)
  600 FORMAT (8(1X,1PE9.2))
      CALL FIT(L2,CARRIV,CURMIN,CURMAX,J,CPH,PHI,N,GAMMA)
 4000 CONTINUE
  360 SECPRN(J)=SECPRN(J)-CARRIV
  400 CONTINUE
      RETURN
      END
      SUBROUTINE GUN
C*********************************************************************
C PURPOSE: TO COMPUTE THE CURRENT DUE TO THE EMISSION OF A GUN
C*********************************************************************
      COMMON/COUR/CION(144),CELEC(144),CSEC(144),CSEC1(144),CHOTO(144)
     1 ,CPHOT1(144),CBSCAT(144),CBSCA1(144),SECPRN(144),SECPR1(144),
     2 PHI(144),COUN(144)
      RETURN
      END
      SUBROUTINE FIT(L2,CARRIV,CURMIN,CURMAX,J,PHOO,PHI,N,GAMMA)
C*******************************************************************
C THIS SUBROUTINE CALCULATES SECONDARY CURRENT FROM OTHER POINTS.
C IT IS USED IN THE COMPUTATION OF PHOTOELECTRONS,SECONDARIES,AND
C BACKSCATTERS ELECTRONS.
C
C PURPOSE
C
C TO COMPUTE THE CURRENT DENSITY=SUM OVER I(VELOCITY) AND SUM OVER L
C (ANGLE) OF (A+B*ANGLE+C*VELOCITY+D*ANGLE*VELOCITY)*EXP(-VELOCITY**2)*
C (VELOCITY**2)*SIN(ANGLE)*D(ANGLE)*D(VELOCITY)
C TACKING INTO ACCOUNT THE FACT THAT FOR CERTAIN INCIDENT ENERGIES
C AND CERTAIN DIRECTIONS, ELECTRON ORBITS DO CONNECT BACK TO THE AMBIENT
C PLASMA.
C DESCRIPTION OF THE VARIABLES
C       CARRIV:TOTAL ARRIVING SECONDARY CURRENT AT A GIVEN GRID POINT
C       CURENT:ARRIVING CURRENT FOR ONE E1EMENTARY CELL IN VENOCITY SPACE
C       PHO(I,L): WEIGHTING FACTORS IN CURRENT INTEGRATION
C       A,B,C,D: COEFFICIENTS IN BILINEAR INTERPOLATION
C       ANG: SAMPLING ANGLE IN VELOCITY SPACE.ANGLES ARE SAMPLED FROM 0 T
C       ZERO TO PI
C       VI: SAMPLING IN KINETIC VELOCITY
C       NLEV: NUMBER OF ENERGY LEVELS USED IN THE COMPUTATION
C       L2: NUMBER OF ANGLES IN VELOCITY SPACE USED IN THE COMPUTATION
C
      INTEGER PERSEC,SECIND,PERPAN,PANIND
      COMMON PSAVE(144,10),CSAVE(144,10),PSECTR(144),CSECTR(144),
     1DELEE(20),DELMAX(144),EMAX(144),ETA(20),BSCAT1(144),
     2BSCAT2(144),BSCAT3(144),PHO(20,40),
     1ANG(40),CPHOMX(144),SUMANG,ANGO,BOUNDL(20,40),BOUNDI(20,40),
     2ANGL(20,40),VII(20,40),ICHECK(20,40),RATIO(13),IROOT(144),
     3IFAULT(14),JLOCAL(144),ICON(144),
     1VI(20),EV(8),NLEV,NTIMES,ILEV,TIMIN(20),TIMAX(20),VIMUM(20),
     2A(144),B(144),C(144),BB(144),BW(144),TW(144),DW(144),
     3EW(144),TWOCOS(700),
     1IFLG,NPOINT,N,IGEOM,ISYM,IPOIS,NSECT,NPEROD,MPEROD,IDINY,
     2GAMMA,DVBX(70,48),DVBY(70,48),Y(144),DELTAX,DELTAY,
     3YS(64,48),XS(64),XSS(70),
     1VARBLE(4),T,DELT,SBOUND,NDIM,POT,H,J1,RPI,RPI2,RPIHAF,SQTPI,
     2SAY,TEMPE1,TEMPE2,DENE1,DENE2,DENI1,DENI2,TEMPI1,TEMPI2,
     3TEMPR,LUN3,LUN4,LUN5,LUN6,LUN7,LUN8,LUN9,LUN10,
     1IBTL(144),JBTL(144),PERSEC(144),SECIND(144),IPANEL(144),
     2PERPAN(144,4),PANIND(144),
     1CION(144),CELEC(144),CSEC(144),CSEC1(144),CPHOTO(144),
     2CPHOT1(144),CBSCAT(144),CBSCA1(144),SECPRN(144),SECPR1(144),
     3PHI(144),COUN(144),
     1KASE,JONL,BPHI,MANG,RAD,PBOUND,OMEGA,ALPHA,BO,JP,IS,ECS,
     2EXPVI2(20),EFTVI(20),IERROR,TETMIN,TETMAX,BETA1,BETA2,JLESS,
     3L2,CARRIV,CPH,XI,CSECEL,CSECE1,CS,CB,CS1,CB1
C
      DATA KSTEP/5/
C
      DO 390 I=1,NLEV
      DO 340 L=1,L2
      IF(I.GT.1)GO TO 320
      IF(PHI(J)/GAMMA.LT.0.0) GO TO 340
      IF(PHO(I,L).EQ.0.0.AND.PHO(I,L+1).EQ.0.0) GO TO 340
      DENOM=VI(1)*(ANG(L)-ANG(L+1))
      A=PHOO
      B=0.0
      C=(PHO(1,L+1)*ANG(  L)-PHO(1,L)*ANG(  L+1)+PHOO*(ANG(  L+1)-
     1 ANG(  L)))/DENOM
      D=(PHO(1,L)-PHO(1,L+1))/DENOM
      WPRIME=(A+B*ANG(L))*COS(ANG(L))-(A+B*ANG(L+1))*
     1 COS(ANG(L+1))+B*(SIN(ANG(L+1))-SIN(ANG(L)))
      CURENT=WPRIME*(-0.5*VI(1)*EXPVI2(1)+0.25*SQTPI*EFT(1))
      UPRIME=0.5*(1.0-EXPVI2(1)*(1.0+VI(1)**2))
      CURENT=CURENT+UPRIME*(C*(COS(ANG(L))-COS(ANG(L+1)))+D*(ANG(L)*
     1 COS(ANG(L))-ANG(L+1)*COS(ANG(L+1))+SIN(ANG(L+1))-SIN(ANG(L))))
      CARRIV=CARRIV+CURENT
C
      GO TO 340
C
  320 CONTINUE
      BETA=VI(I-1)**2+PHI(J)/GAMMA
      IF(BETA.LT.0.0) GO TO 340
      DENOM=(ANG(L)-ANG(L+1))*(VI(I)-VI(I-1))
      IF(PHO(I-1,L+1).EQ.0..AND.PHO(I,L+1).EQ.0..AND.PHO(I-1,L).EQ.0.
     1 .AND.PHO(I,L).EQ.0.) GO TO 340
C
C CASE 1
      IF (PHO(I,L+1).EQ.0.0 .AND.PHO(I-1,L+1).NE.0.0.AND.PHO(I-1,L).
     1 NE.0.0.AND.PHO(I,L).NE.0.0.AND.ICHECK(I,L+1).EQ.0)
     2 GO TO 400
C
C CASE 2
      IF (PHO(I,L+1).EQ.0.0.AND.PHO(I,L).EQ.0.0.AND.PHO(I-1,L).NE.0.0.
     1 AND.PHO(I-1,L+1).NE.0.0.AND.ICHECK(I,L+1).EQ.0.AND.
     2 ICHECK(I,L).EQ.0.0) GO TO 410
C
C CASE 3
      IF (PHO(I,L+1).EQ.0.0.AND.PHO(I,L).EQ.0.0.AND.PHO(I-1,L+1).EQ.0.0
     1.AND.PHO(I-1,L).NE.0.0.AND.ICHECK(I,L+1).EQ.0.AND.
     2 ICHECK(I,L).EQ.0.AND.ICHECK(I-1,L+1).EQ.0) GO TO 420
C
C CASE 4
      IF (PHO(I,L+1).EQ.0.0.AND.PHO(I,L).EQ.0.0.AND.PHO(I-1,L+1).NE.0.0.
     1 AND.PHO(I-1,L).EQ.0.0.AND.ICHECK(I,L+1).EQ.0.AND.
     2 ICHECK(I,L).EQ.0.AND.ICHECK(I-1,L).EQ.0) GO TO 430
C
C CASE 5
      IF (PHO(I,L+1).NE.0.0.AND.PHO(I,L).EQ.0.0.AND.PHO(I-1,L+1).NE.0.0.
     1 AND.PHO(I-1,L).NE.0.0.AND.ICHECK(I,L).EQ.0) GO TO
     2 440
C
C CASE 6
      IF (PHO(I,L+1).EQ.0.0.AND.PHO(I-1,L+1).EQ.0.0.AND.PHO(I,L).NE.0.0.
     1 AND.PHO(I-1,L).NE.0.0.AND.ICHECK(I,L+1).EQ.0.AND.
     2 ICHECK(I-1,L+1).EQ.0) GO TO 450
C
C CASE 7
      IF (PHO(I,L+1).NE.0.0.AND.PHO(I-1,L+1).NE.0.0.AND.PHO(I,L).EQ.0.0.
     1 AND.PHO(I-1,L).EQ.0.AND.ICHECK(I,L).EQ.0.AND.
     2 ICHECK(I-1,L).EQ.0) GO TO 460
C
      IF(PHO(I-1,L).NE.0.0.AND.PHO(I-1,L+1).NE.0.0.AND.PHO(I,L).
     1 NE.0.0.AND.PHO(I,L+1).NE.0.0) GO TO 9999
      GO TO 340
C START COMPUTATION FOR THE REGULAR CASE
 9999 A=(PHO(I,L)*ANG(  L+1)*VI(I-1)-PHO(I-1,L)*ANG(  L+1)*VI(I)-PHO(I,
     1L+1)*ANG(  L)*VI(I-1)+PHO(I-1,L+1)*ANG(  L)*VI(I))/DENOM
      B=(VI(I)*(PHO(I-1,L)-PHO(I-1,L+1))-VI(I-1)*(PHO(I,L)-PHO(I,L+1)))
     1/DENOM
      C=(ANG(  L)*(PHO(I,L+1)-PHO(I-1,L+1))-ANG(  L+1)*(PHO(I,L)-PHO(
     1I-1,L)))/DENOM
      D=(PHO(I,L)-PHO(I-1,L)-PHO(I,L+1)+PHO(I-1,L+1))/DENOM
      WPRIME=(A+B*ANG(L))*COS(ANG(L))-(A+B*ANG(L+1))*
     1 COS(ANG(L+1))+B*(SIN(ANG(L+1))-SIN(ANG(L)))
      CURENT=WPRIME*(0.5*(VI(I-1)*EXPVI2(I-1)-VI(I)*EXPVI2(I))
     1 +0.25*SQTPI*(EFTVI(I)-EFT(I-1)))
      UPRIME=0.5*(EXPVI2(I-1)*(1.0+VI(I-1)**2)-EXPVI2(I)*(1.0+
     1 VI(I)**2))
      CURENT=CURENT+UPRIME*(C*(COS(ANG(L))-COS(ANG(L+1)))+D*(ANG(L)*COS(
     1 ANG(L))-ANG(L+1)*COS(ANG(L+1))+SIN(ANG(L+1))-SIN(ANG(L))))
      CARRIV=CARRIV+CURENT
      GO TO 340
C
C START COMPUTATIONS FOR THE NONREGULAR CASE
C
C----CASE NB 1-----
  400 CONTINUE
      PA=BOUNDI(I,L+1)
      PB=BOUNDL(I,L+1)
      PD=(PHO(I-1,L+1)-PHO(I-1,L))/(ANG(L+1)-ANG(L))*
     1 (ANGL(I,L+1)-ANG(L))+PHO(I-1,L)
C
C REGION 1
C----------
      D1=(PB-PD-PHO(I,L)+PHO(I-1,L))/((ANGL(I,L+1)-ANG(L))*(VI(I)-
     1 VI(I-1)))
      C1=(ANGL(I,L+1)*(PHO(I,L)-PHO(I-1,L))-ANG(L)*(PB-PD))/
     1 ((ANGL(I,L+1)-ANG(L))*(VI(I)-VI(I-1)))
      B1=(PB-PHO(I,L))/(ANGL(I,L+1)-ANG(L))-D1*VI(I)
      A1=PB-B1*ANGL(I,L+1)-C1*VI(I)-D1*ANGL(I,L+1)*VI(I)
C
C REGION 2
C----------
      VA=VII(I,L+1)
      ANGB=ANGL(I,L+1)
      D2=((PB-PD)*(VA-VI(I-1))-(PA-PHO(I-1,L+1))*(VI(I)-VI(I-1)))/
     1 ((VA-VI(I-1))*(VI(I)-VI(I-1))*(ANGB-ANG(L+1)))
      C2=((PB-PD)*ANG(L+1)*(VA-VI(I-1))-(PA-PHO(I-1,L+1))*ANGB*(VI(I)-
     1 VI(I-1)))/((VA-VI(I-1))*(VI(I)-VI(I-1))*(ANG(L+1)-ANGB))
      B2=(PB-PHO(I-1,L+1))/(ANGB-ANG(L+1))-D2*VI(I-1)
      A2=PHO(I-1,L+1)-B2*ANG(L+1)-C2*VI(I-1)-D2*VI(I-1)*ANG(L+1)
C
C
       WPRIME=(A1+B1*ANG(L))*COS(ANG(L))-(A1+B1*ANGB)*COS(ANGB)+
     1 B1*(SIN(ANGB)-SIN(ANG(L)))
      CURENT=WPRIME*(-0.5*(VI(I-1)*EXPVI2(I-1)-VI(I)*EXPVI2(I)
     1)+0.5*SQTPI*(EFTVI(I)-EFT(I-1)))
      UPRIME=0.5*(EXPVI2(I-1)*(1.0+VI(I-1)**2)-EXPVI2(I)*
     1 (1.0+VI(I)**2))
      CURENT=CURENT+UPRIME*(C1*(COS(ANG(L))-COS(ANGB))+D1*(ANG(L)*COS(AN
     1 G(L))
     2 -ANGB*COS(ANGB)+SIN(ANGB)-SIN(ANG(L))))
C
      STEP=(ANG(L+1)-ANGB)/FLOAT(KSTEP)
      STEPA=ANGB
      STEPC=(VII(I,L+1)-VI(I))/FLOAT(KSTEP)
      STEPD=VI(I)
C
      DO 401 K=1,KSTEP
      STEPB=STEPA+STEP
      WPRIME=(A2+B2*STEPA)*COS(STEPA)-(A2+B2*STEPB)*COS(STEPB)+
     1 B2*(SIN(STEPB)-SIN(STEPA))
      CURENT=CURENT+WPRIME*(0.5*(VI(I-1)*EXPVI2(I-1)-STEPD*EXP(-STE
     1**2)) +0.25*SQTPI*(EFT(STEPD)-EFT(I-1)))
      UPRIME=0.5*(EXPVI2(I-1)*(1.0+VI(I-1)**2)-EXP(-STEPD**2)*
     1 (1.0+STEPD**2))
      CURENT=CURENT+UPRIME*(C2*(COS(STEPA)-COS(TSEPB))+
     1 B2*(STEPA*COS(STEPA)-STEPB*COS(STEPB)+SIN(STEPB)-SIN(STEPA)))
      STEPD=STEPD+STEPC
      STEPA=STEPB
  401 CONTINUE
      CARRIV=CARRIV+CURENT
      CUR1=CURENT
  901 FORMAT(* CUR1*,3I3,1PE10.3)
      WRITE(9,901) J,I,L,CUR1
C
      GO TO 340
C
C----CASE NB 2-----
  410 D=((BOUNDI(I,L+1)-PHO(I-1,L+1))*(VII(I,L)-VI(I-1))-(VII(I,L+1)-
     1 VI(I-1))*(BOUNDI(I,L)-PHO(I-1,L)))/(ANG(L)*(VI(I-1)-VII(I,L))*
     1 (VII(I,L+1)-VI(I-1))+ANG(L+1)*(VII(I,L+1)-VI(I-1))*(VII(I,L)-
     1 VI(I-1)))
      C=(BOUNDI(I,L)-PHO(I-1,L)+D*ANG(L)*(VI(I-1)-VII(I,L)))/(VII(I,L)
     1 -VI(I-1))
      B=(PHO(I-1,L)-PHO(I-1,L+1)-D*VI(I-1)*(ANG(L)-ANG(L+1)))/(ANG(L)-
     1 ANG(L+1))
      A=PHO(I-1,L+1)-B*ANG(L+1)-C*VI(I-1)-D*ANG(L+1)*VI(I-1)
      CURENT=0.0
      STEP=(ANG(L+1)-ANG(L))/FLOAT(KSTEP)
```

```
      STEPA=ANG(L)
      STEPC=(VII(I,L+1)-VII(I,L))/FLOAT(KSTEP)
      STEPD=VII(I,L)
C
      DO 411 K=1,KSTEP
      STEPB=STEPA+STEP
      WPRIME=(A+B*STEPA)*COS(STEPA)-(A+B*STEPB)*COS(STEPB)+
     1 B*(SIN(STEPB)-SIN(STEPA))
      CURENT=CURENT+WPRIME*(0.5*(VI(I-1)*EXPVI2(I-1)-STEPD*EXP
     1 (-STEPD**2))+0.25*SQTPI*(EFT(STEPD)-EFT(I-1)))
      UPRIME=0.5*(EXPVI2(I-1)*(1.0+VI(I-1)**2)-EXP(-STEPD**2)*
     1 (1.0+STEPD**2))
      CURENT=CURENT+UPRIME*(C*(COS(STEPA)-COS(STEPB))+D*(STEPA*COS(STEPA
     1) -STEPB*COS(STEPB)+SIN(STEPB)-SIN(STEPA)))
      STEPA=STEPB
      STEPD=STEPD+STEPC
  411 CONTINUE
      CARRIV=CARRIV+CURENT
      CUR2=CURENT
      IF (J.EQ.1) WRITE(9,902) J,I,L,CUR2
  902 FORMAT(* CUR2*,3I3,1PE10.3)
C
      GO TO 340
C
C----CASE NB 3----
  420 D=((BOUNDI(I,L)-PHO(I-1,L))*(VII(I,L)-VI(I-1))-
     1 (BOUNDI(I,L)-BOUNDL(I-1,L+1))*VII(I,L))/ (VII(I,L)*ANG(L)*
     2 (VII(I,L)-VI(I-1))+(VII(I,L)*ANG(L)-VI(I-1)*ANGL(I-1,L+1))*VII
     3 (I,L))
      C=(BOUNDI(I,L)-PHO(I-1,L)-D*VII(I,L)*ANG(L))/VII(I,L)
      B=0.0
      A=PHO(I-1,L)
C
      STEP=(ANGL(I-1,L+1)-ANG(L))/FLOAT(KSTEP)
      STEPA=ANG(L)
      CURENT=0.0
      STEPC=(VI(I-1)-VII(I,L))/FLOAT(KSTEP)
      STEPD=VII(I,L)
C
      DO 421 K=1,KSTEP
      STEPB=STEPA+STEP
      WPRIME=(A+B*STEPA)*COS(STEPA)-(A+B*STEPB)*COS(STEPB)+
     1 B*(SIN(STEPB)-SIN(STEPA))
      CURENT=CURENT+WPRIME*(0.5*(VI(I-1)*EXPVI2(I-1)-STEPD*
     1 EXP(-STEPD**2))+0.25*SQTPI*(EFT(STEPD)-EFT(I-1)))
      UPRIME=0.5*(EXPVI2(I-1)*(1.0+VI(I-1)**2)-EXP(-STEPD**2)*
     1 (1.0+STEPD**2))
      CURENT=CURENT+UPRIME*(C*(COS(STEPA)-COS(STEPB))+D*(STEPA*
     1 COS(STEPA)-STEPB*COS(STEPB)+SIN(STEPB)-SIN(STEPA)))
      STEPA=STEPB
      STEPD=STEPD+STEPC
  421 CONTINUE
      CARRIV=CARRIV+CURENT
      CUR3=CURENT
      WRITE(9,903) J,I,L,CUR3
  903 FORMAT(* CUR3*,3I3,1PE10.3)
C
      GO TO 340
C
C----CASE NB 4----
C
  430 D=((BOUNDI(I,L+1)-PHO(I-1,L+1))*(VII(I,L+1)-VI(I-1))-
     1 (BOUNDI(I,L+1)-BOUNDL(I-1,L))*VII(I,L+1))/(VII(I,L+1)*ANG(L+1)*
     1 *(VII(I,L+1)-VI(I-1))+(VII(I,L+1)*ANG(L+1)-VI(I-1)*ANGL(I-1,L))
     3 *VII(I,L+1))
      C=(BOUNDI(I,L+1)-PHO(I-1,L+1)-D*VII(I,L+1)*ANG(L+1))/VII(I,L+1)
      B=0.0
      A=PHO(I 1,L+1)
      STEP=(ANG(L+1)-ANGL(I-1,L))/FLOAT(KSTEP)
      STEPA=ANGL(I-1,L)
      CURENT=0.0
      STEPC=(VII(I,L+1)-VI(I-1))/FLOAT(KSTEP)
      STEPD=VI(I-1)
C
      DO 431 K=1,KSTEP
      STEPB=STEPA+STEP
      WPRIME=(A+B*STEPA)*COS(STEPA)-(A+B*STEPB)*COS(STEPB)+
     1 B*(SIN(STEPB)-SIN(STEPA))
      CURENT=CURENT+WPRIME*(0.5*(VI(I-1)*EXPVI2(I-1)-STEPD*EXP(
     1 -STEPD**2))+0.25*SQTPI*(EFT(STEPD)-EFT(I-1)))
      UPRIME=0.5*(EXPVI2(I-1)*(1.0+VI(I-1)**2)-EXP(-STEPD**2)
     1 *(1.0+STEPD**2))
      CURENT=CURENT+UPRIME*(C*(COS(STEPA)-COS(STEPB))+D*(STEPA*
     1 COS(STEPA)-STEPB*COS(STEPB)+SIN(STEPB;-SIN(STEPA)))
      STEPA=STEPB
      STEPD=STEPD+STEPC
  431 CONTINUE
C
      CARRIV=CARRIV+CURENT
      CUR4=CURENT
      IF(J.EQ.1) WRITE(9,904) J,I,L,CUR4
  904 FORMAT ( * CUR4*,3I3,1PE10.3)
C
      GO TO 340
C
C----CASE NB 5----
  440 CONTINUE
      PD=(PHO(I-1,L+1)-PHO(I-1,L))/(ANG(L+1)-ANG(L))*(ANGL(I,L)-ANG(L))
     1 +PHO(I-1,L)
      D1=(BOUNDL(I,L)-PD-PHO(I,L+1)+PHO(I-1,L+1))/((VI(I)-VI(I-1))*
     1 (ANGL(I,L)-ANG(L+1)))
      C1=(BOUNDL(I,L)-PD-D1*ANGL(I,L)*(VI(I)-VI(I-1)))/(VI(I)-VI(I-1))
      B1=(PHO(I,L+1)-BOUNDL(I,L)-D1*(ANG(L+1)*VI(I)-ANGL(I,L)*VI(I)))/
     1 (ANG(L+1)-ANGL(I,L))
      CURENT=0.0
      A1=PHO(I,L+1)-B1*ANG(L+1)-C1*VI(I)-D1*ANG(L+1)*VI(I)
C
      WPRIME=(A1+B1*ANGL(I,L))*COS(ANGL(I,L))-(A1+B1*ANG(L+1))*
     1 COS(ANG(L+1))+B1*(SIN(ANG(L+1))-SIN(ANGL(I,L)))
      CURENT=CURENT+WPRIME*(0.5*(VI(I-1)*EXPVI2(I-1)-VI(I)*
     1 EXPVI2(I))+0.25*SQTPI*(EFTVI(I)-EFT(I-1)))
      UPRIME=0.5*(EXPVI2(I-1)*(1.0+VI(I-1)**2)-EXPVI2(I)*
     1 (1.0+VI(I)**2))
      CURENT=CURENT+UPRIME*(C1*(COS(ANGL(I,L))-COS(ANG(L+1)))+D1*
     1 (ANGL(I,L)*COS(ANGL(I,L))-ANG(L+1)*COS(ANG(L+1))+
     1 SIN(ANG(L+1))-SIN(ANGL(I,L))))
 4400 CONTINUE
       IF(ANGL(I,L).EQ.ANG(L)) GO TO 4401
      D2=((BOUNDL(I,L)-PB)*(VI(I-1)-VII(I,L))-PHO(I-1,L)-BOUNDI(I,L)*
     1 (VI(I)-VI(I-1)))/(ANG(L)*(VI(I-1)-VII(I,L))*(VI(I)-VI(I-1))+
     2 ANGL(I,L)*(VI(I)-VI(I-1))*(VI(I-1)-VII(I,L)))
      C2=(PHO(I-1,L)-BOUNDI(I,L)-D2*ANG(L)*(VI(I-1)-VII(I,L)))/(VI(I-1)
     1 -VII(I,L))
      B2=(PD-PHO(I-1,L)-D2*(ANGL(I,L)*VI(I-1)-ANG(L)*VI(I-1)))/(ANGL(I,
     1 L)-ANG(L))
      A2=PHO(I-1,L)-B2*ANG(L)-C2*VI(I-1)-D2*ANG(L)*VI(I-1)
      STEP=(ANGL(I,L)-ANG(L))/FLOAT(KSTEP)
      STEPA=ANG(L)
      STEPC=(VI(I)-VII(I,L))/FLOAT(KSTEP)
      STEPD=VII(I,L)
C
      DO 441 K=1,KSTEP
      STEPB=STEPA+STEP
      WPRIME=(A2+B2*STEPA)*COS(STEPA)-(A2+B2*STEPB)*COS(STEPB)+
     1 B2*(SIN(STEPB)-SIN(STEPA))
      CURENT=CURENT+WPRIME*(0.5*(VI(I-1)*EXPVI2(I-1)-STEPD*EXP
     1 (-STEPD**2))+0.25*SQTPI*(EFT(STEPD)-EFT(I-1)))
      UPRIME=0.5*(EXPVI2(I-1)*(1.0+VI(I-1)**2)-EXP(-STEPD**2)*
     1 (1.0+STEPD**2))
      CURENT=CURENT+UPRIME*(C2*(COS(STEPA)-COS(STEPB))+D2*(STEPA*
     1 COS(STEPA)-STEPB*COS(STEPB)+SIN(STEPB)-SIN(STEPA)))
      STEPA=STEPB
      STEPD=STEPD+STEPC
  441 CONTINUE
C
 4401 CARRIV=CARRIV+CURENT
      CUR5=CURENT
      IF(J.EQ.1) WRITE(9,905) J,I,L,CUR5
  905 FORMAT( * CUR5*,3I3,1PE10.3)
      GO TO 340
C
  450 CONTINUE
C----CASE NB 6----
C
      D=((PHO(I-1,L+1)-BOUNDL(I-1,L))*(ANG(L+1)-ANGL(I,L))
     1 -(PHO(I,L+1)-BOUNDL(I,L))*(ANG(L+1)-ANGL(I-1,L)))/
     2 ((ANG(L+1)-ANGL(I-1,L))*(ANG(L+1)-ANGL(I,L))*
     3 (VI(I-1)-VI(I)))
      C=(PHO(I,L+1)-PHO(I-1,L+1)-D*ANG(L+1)*(VI(I)-VI(I-1)))/
     1 (VI(I)-VI(I-1))
      B=(PHO(I,L+1)-BOUNDL(I,L)-D*VI(I)*(ANG(L+1)-ANGL(I,L)))/
     1 (ANG(L+1)-ANGL(I,L))
      A=PHO(I,L+1)-B*ANG(L+1)-C*VI(I)-D*ANG(L+1)*VI(I)
      STEPA=0.5*(ANGL(I,L)+ANGL(I-1,L))
      WPRIME=(A+B*STEPA)*COS(STEPA)-(A+B*ANG(L+1))*COS(ANG(L+1))+
     1 B*(SIN(ANG(L+1))-SIN(STEPA))
      CURENT=WPRIME*(0.5*(VI(I-1)*EXPVI2(I-1)-VI(I)*EXP(-VI(I)**2
     1))+0.25*SQTPI*(EFTVI(I)-EFT(I-1)))
      UPRIME=0.5*(EXPVI2(I-1)*(1.0+VI(I-1)**2)-EXPVI2(I)*
     1 (1.0+VI(I)**2))
      CURENT=CURENT+UPRIME*(C*(COS(STEPA)-COS(ANG(L+1)))+D*(STEPA*
     1 COS(STEPA)-ANG(L+1)*COS(ANG(L+1))+SIN(ANG(L+1))-SIN(STEPA)))
      CARRIV=CARRIV+CURENT
      CUR6=CURENT
      IF(J.EQ.1) WRITE(9,906) J,I,L,CUR6
  906 FORMAT(* CUR6*,3I3,1PE10.3)
C
      GO TO 340
C
  460 CONTINUE
C
C----CASE NB 7----
C
      PD=BOUNDL(I-1,L+1)
      PB=BOUNDL(I,L+1)
      D=((PHO(I-1,L)-PD)*(ANG(L)-ANGL(I,L+1))-(PHO(I,L)-PB)*
     1 (ANG(L)-ANGL(I-1,L+1)))/((VI(I-1)-VI(I))*(ANG(L)-ANGL(I-1,L+1))*
     2 (ANG(L)-ANGL(I,L+1)))
      C=(PHO(I,L)-PHO(I-1,L) -D*ANG(L)*(VI(I)-VI(I-1)))/(VI(I)-VI(I-1))
      B=(PHO(I,L)-PB-D*VI(I)*(ANG(L)-ANGL(I,L+1)))/(ANG(L)-ANGL(I,L+1))
      A=PHO(I,L)-B*ANG(L)-C*VI(I)-D*VI(I)*ANG(L)
      STEPA=0.5*(ANGL(I,L+1)+ANGL(I-1,L+1))
      WPRIME=(A+B*ANG(L))*COS(ANG(L))-(A+B*STEPA)*COS(STEPA)+
     1 B*(SIN(STEPA)-SIN(ANG(L)))
      CURENT=WPRIME*(0.5*(VI(I-1)*EXPVI2(I-1)-VI(I)*EXPVI2(I))
     1+0.25*SQTPI*(EFTVI(I)-EFT(I-1)))
      UPRIME=0.5*(EXPVI2(I-1)*(1.0+VI(I-1)**2)-EXPVI2(I)*
     1 (1.0+VI(I)**2))
      CURENT=CURENT+UPRIME*(C*(COS(ANG(L))-COS(STEPA))+D*(ANG(L)*
     1 COS(ANG(L))-STEPA*COS(STEPA)+SIN(STEPA)-SIN(ANG(L))))
      CARRIV=CARRIV+CURENT
      CUR7=CURENT
      IF(J.EQ.1) WRITE(9,907) J,I,L,CUR7
  907 FORMAT(* CUR7*,3I3,1PE10.3)
  340 CONTINUE
  390 CONTINUE
C
C COMPUTATION FROM THE CUTOFF BOUNDARY TO THE NEAREST LEVEL
      IF(PHI(J).LE.0.0) RETURN
      I1=0
      DO 500 I=1,NLEV
      BETA=VI(I)**2+PHI(J)/GAMMA
      IF(BETA.LT.0.0) I1=I
  500 CONTINUE
      IF (I1.EQ.NLEV) RETURN
      I=I1+1
      V1=SQRT(-PHI(J)/GAMMA)
       IF(VI(I).EQ.V1) RETURN
      CURENT=0.0
      DO 501 L=1,L2
      IF(PHO(I,L).EQ.0.0.AND.PHO(I,L+1).EQ.0.0) GO TO 501
      DENOM=(ANG(L)-ANG(L+1))*(VI(I)-V1)
      A=(PHO(I,L)*ANG(L+1)*V1-PHOO*ANG(L+1)*VI(I)-PHO(I,L+1)*ANG(L)
     1 *V1+PHOO*ANG(L)*VI(I))/DENOM
      B=-V1*(PHO(I,L)-PHO(I,L+1))/DENOM
      C=(ANG(L)*(PHO(I,L+1)-PHOO)-ANG(L+1)*(PHO(I,L)-PHOO))/DENOM
      D=(PHO(I,L)-PHO(I,L+1))/DENOM
      WPRIME=(A+B*ANG(L))*COS(ANG(L))-(A+B*ANG(L+1))*COS(ANG(L+1))
     1 +B*(SIN(ANG(L+1))-SIN(ANG(L)))
      CURENT=CURENT+WPRIME*(0.5*(V1*EXP(-V1**2)-VI(I)*EXPVI2(I))+
     1 0.25*SQTPI*(EFTVI(I)-EFT(V1)))
      UPRIME=0.5*(EXP(-V1**2)*(1.0+V1**2)-EXPVI2(I)*(1.0+VI(I)**2))
      CURENT=CURENT+UPRIME*(C*(COS(ANG(L))-COS(ANG(L+1)))+D*(ANG(L)*
     1 COS(ANG(L))-ANG(L+1)*COS(ANG(L+1))+SIN(ANG(L+1))-SIN(ANG(L))))
  501 CONTINUE
C
      CARRIV=CARRIV+CURENT
C
      RETURN
       END
*ENA(BLK1,0)
      SUBROUTINE DENORD(TETMIN,TETMAX,BETA1,BETA2,I,J,JLESS,
     1 IGEOM,OMEGA,SO)
C
C
C
C
C
C
C
C
C
C
C*************************************************************************
C PURPOSE OF THE ROUTINE
C TO COMPUTE ACCEPTANCE ANGLES
C DESCRIPTION OF THE VARIABLES
C       ACCMIN IS CONVERGENCE CRITERIA FOR ACCEPTANCE ANGLE
C       DELMIN IS INCREMENT IN ANGLE WHEN WE SWEEP FORM 0 TO PI
C
C*************************************************************************
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      INTEGER PERSEC,SECIND,PERPAN,PANIND
      COMMON/BLK1/
```

```
1      PSAVE(144,10),CSAVE(144,10),PSECTR(144),CSECTR(144),
     1DELEE(20),DELMAX(144),EMAX(144),ETA(20),BSCAT1(144),
     2BSCAT2(144),BSCAT3(144),PHO(20,40),
     1ANG(40),CPHOMX(144),SUNANG,ANGO,BOUNDL(20,40),BOUNDI(20,40),
     2ANGL(20,40),VI1(20,40),ICHECK(20,40),RATIO(13),IROOT(144),
     3IFAULT(14),JLOCAL(144),ICON(144),
     1VI(20),EV(8),NLEV,NTIMES,ILEV,TIMIN(20),TIMAX(20),VINUM(20),
     2A(144),B(144),C(144),BB(144),BW(144),TW(144),DW(144),
     3EW(144),TWOCOS(700),
     1IFLG,NPOINT,M,IGEOM,ISYM,IPOIS,NSECT,NPEROD,MPEROD,IDIMY,
     2GAMMA,DVDX(70,48),DVDY(70,48),Y(144),DELTAX,DELTAY,
     3YS(66,48),XS(66),XSS(70),
     1VARBLE(4),T,DELT,SBOUND,NDIM,POT,H,J1,RPI,RPI2,RPIHAF,SQTPI,
     2SAY,TEMPE1,TEMPE2,DENE1,DENE2,DENI1,DENI2,TEMPI1,TEMPI2,
     3TEMPR,LUN3,LUN4,LUN5,LUN6,LUN7,LUN8,LUN9,LUN10,
     1IDTL(144),JDTL(144),PERSEC(144),SECIND(144),IPANEL(144),
     2PERPAN(144,4),PANIND(144),
     1CION(144),CELEC(144),CSEC(144),CSEC1(144),CPHOTO(144),
     2CPHOT1(144),CBSCAT(144),CBSCA1(144),SECPRN(144),SECPR1(144),
     3PHI(144),CGUN(144),
     1KASE,JOML,DPHI,NANG,RAD,PBOUND,OMEGA,ALPHA,SO,JP,IS,ECS,
     2EXPVI2(20),EFTVI(20),IERROR,TETMIN,TETMAX,BETA1,BETA2,JLESS,
     3L2,CARRIV,CPH,XI,CSECEL,CSECE1,CS,CB,CS1,CB1
      DATA DELMIN,ACCMIN/18.,1.E-2/
C
C
C
  160 JLESS=0
      JJJJ=0
      KKKK=0
      LLLL=0
      T1OLD=TETMIN
      T2OLD=TETMAX
      VEL=SQRT(BETA2)
C
C
C
C CALCULATE THE ANGLES ZERO AND PIE AND RPIHAF
C ANGLE ZERO
      TETMIN=0.
      TETMAX=RPI
      TETMED=RPIHAF
C
      DBETA=-DVDX(1,J)/(2.0*BETA2*GAMMA)+1.0
      IF(DBETA.LT.0.0) GO TO 1000
      VARBLE(1)=SO
      VARBLE(2)=0.0
      VARBLE(3)=Y(J)
      VARBLE(4)=-VEL
      T=0.
      DELT=H
      CALL ORBIT(SO,BETA1,J)
  501 FORMAT(1X,5F13.5,2I4,3F13.5)
      IF(VARBLE(1).GE.(SBOUND-DELTAX))KKKK=1
C
C
C
C ANGLE PIE
 1000 VARBLE(1)=SO
      VARBLE(2)=0.
      VARBLE(3)=Y(J)
      VARBLE(4)=+VEL
      T=0.
      DELT=H
      DBETA=DVDX(1,J)/(2.0*BETA2*GAMMA)-1.0
      IF(DBETA.GT.0.0) GO TO 2000
      CALL ORBIT(SO,BETA1,J)
      IF(VARBLE(1).GE.(SBOUND-DELTAX))JJJJ=1
C IF PARTICLE ESCAPES AT BOTH ZERO AND PIE, RETURN TO ION
      IF(JJJJ.EQ.1.AND.KKKK.EQ.1)RETURN
C
C
C ANGLE RPI/2
C
 2000 VARBLE(1)=SO
      VARBLE(2)=VEL
      VARBLE(3)=Y(J)
      VARBLE(4)=0.0
      T=0.
      DELT=H
      CALL ORBIT(SO,BETA1,J)
      IF(VARBLE(1).GE.(SBOUND-DELTAX))LLLL=1
      IF(LLLL.NE.1) GO TO 5000
      TETHGH=RPIHAF
      TETLOW=RPIHAF
C
C PARTICLE ESCAPES AT RPIHAF
C FIND MINIMUM ANGLE TETMIN BY BISSECTION
C
      IF(KKKK.EQ.1) GO TO 4000
C
 3000 TETMED=(TETHGH+TETMIN)/2.0
      VARBLE(1)=SO
      VARBLE(2)=VEL*SIN(TETMED)
      VARBLE(3)=Y(J)
      VARBLE(4)=-VEL*COS(TETMED)
      T=0.0
      DELT=H
      CALL ORBIT(SO,BETA1,J)
      IF(VARBLE(1).GE.(SBOUND-DELTAX)) TETHGH=TETMED
      IF(VARBLE(1).LE.SO) TETMIN=TETMED
      IF(TETHGH-TETMIN .GE.ACCMIN) GO TO 3000
C
C FIND MAXIMUM ANGLE TETMAX BY BISSECTION
C
      IF (JJJJ.EQ.1) RETURN
C
 4000 CONTINUE
 4500 TETMED=(TETLOW+TETMAX)/2.0
      VARBLE(1)=SO
      VARBLE(2)=VEL*SIN(TETMED)
      VARBLE(3)=Y(J)
      VARBLE(4)=-VEL*COS(TETMED)
      T=0.0
      DELT=H
      CALL ORBIT(SO,BETA1,J)
      IF(VARBLE(1).GE.(SBOUND-DELTAX)) TETLOW=TETMED
      IF(VARBLE(1).LE.SO) TETMAX=TETMED
      IF(TETMAX-TETLOW.GE.ACCMIN) GO TO 4500
      TETMAX=TETLOW
      RETURN
C
C
C
 5000 CONTINUE
C THIS SECTION ATTEMPTS TO FIND AT LEAST ONE ALLOWED ORBIT WHEN
C ALL KKKK,JJJJ,LLLL=0.
C DO ANGLE SCANING FROM LEFT TO RIGHT EVERY 10 DEGREES
 4999 DO 5001 L=1,17
      TETMED=RPI*FLOAT(L)/DELMIN
      VARBLE(1)=SO
      VARBLE(2)=VEL*SIN(TETMED)
      VARBLE(3)=Y(J)
      VARBLE(4)=-VEL*COS(TETMED)
      T=0.0
      DELT=H
      CALL ORBIT(SO,BETA1,J)
      IF(VARBLE(1).GE.(SBOUND-DELTAX)) GO TO 5002
 5001 CONTINUE
      JLESS=2
      RETURN
C
 5002 TETMIN=TETMED-RPI/DELMIN
      TETHGH=TETMED
      TETLOW=TETMED
      GO TO 3000
C
      END
$EMA(BLK1,0)
      SUBROUTINE ORBIT(SO,BETA1,J)
C FOR MORE EFFICIENT SCHEME SEE BURDEN ?
C
C*********************************************************************
C PURPOSE:
C TO COMPUTE ORBITS NUMERICALLY USING RUNGE-KUTTA TECHNIQUES
C
C*********************************************************************
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      INTEGER PERSEC,SECIND,PERPAN,PANIND
      COMMON/BLK1/
     1      PSAVE(144,10),CSAVE(144,10),PSECTR(144),CSECTR(144),
     1DELEE(20),DELMAX(144),EMAX(144),ETA(20),BSCAT1(144),
     2BSCAT2(144),BSCAT3(144),PHO(20,40),
     1ANG(40),CPHOMX(144),SUNANG,ANGO,BOUNDL(20,40),BOUNDI(20,40),
     2ANGL(20,40),VII(20,40),ICHECK(20,40),RATIO(13),IROOT(144),
     3IFAULT(14),JLOCAL(144),ICON(144),
     1VI(20),EV(8),NLEV,NTIMES,ILEV,TIMIN(20),TIMAX(20),VINUM(20),
     2A(144),B(144),C(144),BB(144),BW(144),TW(144),DW(144),
     3EW(144),TWOCOS(700),
     1IFLG,NPOINT,M,IGEOM,ISYM,IPOIS,NSECT,NPEROD,MPEROD,IDIMY,
     2GAMMA,DVDX(70,48),DVDY(70,48),Y(144),DELTAX,DELTAY,
     3YS(66,48),XS(66),XSS(70),
     1VARBLE(4),T,DELT,SBOUND,NDIM,POT,H,J1,RPI,RPI2,RPIHAF,SQTPI,
     2SAY,TEMPE1,TEMPE2,DENE1,DENE2,DENI1,DENI2,TEMPI1,TEMPI2,
     3TEMPR,LUN3,LUN4,LUN5,LUN6,LUN7,LUN8,LUN9,LUN10,
     1IDTL(144),JDTL(144),PERSEC(144),SECIND(144),IPANEL(144),
     2PERPAN(144,4),PANIND(144),
     1CION(144),CELEC(144),CSEC(144),CSEC1(144),CPHOTO(144),
     2CPHOT1(144),CBSCAT(144),CBSCA1(144),SECPRN(144),SECPR1(144),
     3PHI(144),CGUN(144),
     1KASE,JOML,DPHI,NANG,RAD,PBOUND,OMEGA,ALPHA,SO,JP,IS,ECS,
     2EXPVI2(20),EFTVI(20),IERROR,TETMIN,TETMAX,BETA1,BETA2,JLESS,
     3L2,CARRIV,CPH,XI,CSECEL,CSECE1,CS,CB,CS1,CB1
      DATA TOL/1.0E-5/
      BETA2=BETA1-YS(1,J)/GAMMA
      DELT=H
      IF(BETA2.NE.0.0) DELT=H/SQRT(BETA2)
C COMPUTE TIME TO TURNING POINT AND FIND  BETTER TIME STEP
      TAU=2.0*VARBLE(2)*GAMMA/DVDX(1,J)
      IF(TAU.GT.0.0.AND.TAU/20.0.LT.DELT) DELT=TAU
C
C
      PRNT(5)=0.0
      KDECR=0
C
      IRHO=0
C
C COMPUTE DERIVATIVES
      CALL FCT(DERY,BETA1,IRHO,IBETA1)
C
C SAVE VALUES OF PARAMETERS
   11 DO 100 I=1,NDIM
      SAVE(I)=VARBLE(I)
      SADYDX(I)=DERY(I)
  100 CONTINUE
C
   12 DO 1 I=1,NDIM
      AK1(I)=DELT*DERY(I)
      VARBLE(I)=0.5*AK1(I)+SAVE(I)
    1 CONTINUE
C
      CALL OUTP(SO,PRNT)
      IF(PRNT(5).NE.0.0) RETURN
      CALL FCT(DERY,BETA1,IRHO,IBETA1)
      IF(IBETA1.EQ.1) GO TO 10
C
      DO 2 I=1,NDIM
      AK2(I)=DELT*DERY(I)
      VARBLE(I)=0.5*AK2(I)+SAVE(I)
    2 CONTINUE
C
      CALL OUTP(SO,PRNT)
      IF(PRNT(5).NE.0.0) RETURN
      CALL FCT(DERY,BETA1,IRHO,IBETA1)
      IF(IBETA1.EQ.1) GO TO 10
C
      DO 3 I=1,NDIM
      AK3(I)=DELT*DERY(I)
      VARBLE(I)=AK3(I)+SAVE(I)
    3 CONTINUE
C
      CALL OUTP(SO,PRNT)
      IF(PRNT(5).NE.0.0) RETURN
       CALL FCT(DERY,BETA1,IRHO,IBETA1)
      IF(IBETA1.EQ.1) GO TO 10
C
      DO 4 I=1,NDIM
      AK4(I)=DELT*DERY(I)
    4 CONTINUE
C
C COMPUTE THE NEW VALUES OF THE VARIABLES
      DO 5 I=1,NDIM
      VARBLE(I)=(AK1(I)+2.*(AK2(I)+AK3(I))+AK4(I))/6.0+SAVE(I)
    5 CONTINUE
C
      CALL OUTP(SO,PRNT)
      IF(PRNT(5).NE.0.0) RETURN
      CALL FCT(DERY,BETA1,IRHO,IBETA1)
      IF(IBETA1.EQ.1) GO TO 10
C
C COMPUTE TRUNCATION ERROR
      DO 6 I=1,NDIM
      TE(I)=(AK1(I)+AK4(I)-2.0*AK3(I))/6.0
    6 CONTINUE
C
C COMPUTE RELATIVE ERROR FOR EACH VARIABLE
      INI=0
      INA=0
      DO 7 I=1,NDIM
      RTE(I)=0.0
      IF(VARBLE(I).NE.0.0) RTE(I)=TE(I)/VARBLE(I)
      IF(ABS(RTE(I)).GT.TOL) INI=INI+1
```

```
      IF(ABS(RTE(I)).LE.0.01*TOL) IMA=IMA+1
    7 CONTINUE
C
C TEST THE CONVERGENCE
      IF (IMI.EQ.0) GO TO 9
C THE STEP IS REJECTED
   10 DO 8 I=1,NDIM
      VARBLE(I)=SAVE(I)
      DERY(I)=SADYDX(I)
    8 CONTINUE
      DELT=0.5*DELT
      KDECR=KDECR+1
      GO TO 12
C
C
C
C
C THE STEP IS ACCEPTED
    9 IF(IMA.EQ.NDIM) DELT=2.0*DELT
      IF(IMA.EQ.NDIM) KDECR=0
      T=T+DELT
      GO TO 11
      RETURN
      END
$EMA(BLK1,0)
      SUBROUTINE OUTP(SO,PRNT)
C*************************************************************************
C PURPOSE: TO STOP COMPUTATIONS OF THE ORBITS IF THE PARTICLE REACHES TH
C OUTER BOUNDARY OR GOES BACK TO THE SURFACE
C EXTERNALS : NONE
C*************************************************************************
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      INTEGER PERSEC,SECIND,PERPAN,PANIND
      COMMON/BLK1/
     1      PSAVE(144,10),CSAVE(144,10),PSECTR(144),CSECTR(144),
     1DELEE(20),DELMAX(144),EMAX(144),ETA(20),BSCAT1(144),
     2BSCAT2(144),BSCAT3(144),PHO(20,40),
     1ANG(40),CPHOMX(144),SUNANG,ANGO,BOUNDL(20,40),BOUNDI(20,40),
     2ANGL(20,40),VII(20,40),ICHECK(20,40),RATIO(13),IROOT(144),
     3IFAULT(14),JLOCAL(144),ICON(144),
     1VI(20),EV(8),NLEV,NTIMES,ILEV,TIMIN(20),TIMAX(20),VIMUM(20),
     2A(144),B(144),C(144),BB(144),BW(144),TW(144),DW(144),
     3EW(144),TWOCOS(700),
     1IFLG,NPOINT,M,IGEOM,ISYM,IPOIS,NSECT,MPEROD,MPEROD,IDIMY,
     2GAMMA,DVDX(70,48),DVDY(70,48),Y(144),DELTAX,DELTAY,
     3YS(66,48),XS(66),XSS(70),
     1VARBLE(4),T,DELT,SBOUND,NDIM,POT,N,J1,RPI,RPI2,RPIHAF,SQTPI,
     2SAY,TEMPE1,TEMPE2,DENE1,DENE2,DENI1,DENI2,TEMPI1,TEMPI2,
     3TEMPR,LUN3,LUN4,LUN5,LUN6,LUN7,LUN8,LUN9,LUN10,
     1IDTL(144),JDTL(144),PERSEC(144),SECIND(144),IPANEL(144),
     2PERPAN(144,4),PANIND(144),
     1CION(144),CELEC(144),CSEC(144),CSEC1(144),CPHOTO(144),
     2CPHOT1(144),CBSCAT(144),CBSCA1(144),SECPRN(144),SECPR1(144),
     3PHI(144),COUN(144),
     1KASE,JOML,DPHI,NANO,RAD,PBOUND,OMEGA,ALPHA,SO,JP,IS,ECS,
     2EXPVI2(20),EFTVI(20),IERROR,TETMIN,TETMAX,BETA1,BETA2,JLESS,
     3L2,CARRIV,CPH,XI,CSECEL,CSECE1,CS,CB,CS1,CB1
C
C
      NUMORB=0
      ORB=VARBLE(3)
   60 IF(ORB.LT.0.0) ORB=ORB+RPI2
      IF(ORB.GE.RPI2) ORB=ORB-RPI2
      NUMORB=NUMORB+1
      IF(ORB.LT.0..OR.ORB.GE.RPI2) GO TO 60
      IF(NUMORB.LT.5) GO TO 150
C PARTICULE HAS MADE FIVE ORBITS,STOP COMPUTATIONS
      VARBLE(1)=SBOUND
  150 IF(VARBLE(1).LT.SO.OR.VARBLE(1).GE.(SBOUND-DELTAX))GO TO 300
      RETURN
C
  300 CONTINUE
      PRNT(5)=1.0
      IF(VARBLE(1).GE.(SBOUND-DELTAX)) RETURN
C
C INTERPOLATE ORBIT IN ORDER TO FIND POTENTIAL VALUE AT SURFACE
C
      VX=VARBLE(3)
      VARBLE(3)=(VARBLE(3)-V(3))/(VARBLE(1)-V(1))*(-V(1))+V(3)
      ORB=VARBLE(3)
   70 IF(ORB.LT.0.0) ORB=ORB+RPI2
      IF(ORB.GE.RPI2) ORB=ORB-RPI2
      IF(ORB.LT.0..OR.ORB.GE.RPI2) GO TO 70
      VARBLE(3)=ORB
      DO 80 J=1,N
      IF (ORB.LT.Y(1)) GO TO 90
      IF (ORB.GE.Y(N)) GO TO 95
      IF (ORB.GE.Y(J).AND.ORB.LT.Y(J+1)) GO TO 100
   80 CONTINUE
      GO TO 100
   90 J1=N
      ORB=ORB+RPI2
      GO TO 110
   95 J1=N
      GO TO 110
  100 J1=J
  110 CONTINUE
      J2=J1+1
      IF(J1.EQ.N) J2=1
      ORBB=ORB-Y(J1)
      IF(ORBB.LT.0.0) ORBB=ORBB+RPI2
      POT=(YS(1,J2)-YS(1,J1))/DELTAY*(ORBB)+YS(1,J1)
C
C FIND LOCATION J1 FOR SURFACE PROPERTIES ABOUT PHOTOELECTRONS...ETC
C
      IF(ORBB.GE.0.5*DELTAY) J1=J2
      RETURN
      END
$EMA(BLK1,0)
      SUBROUTINE FCT(DERY,BETA1,IRHO,IBETA1)
C*********************************************************************
C PURPOSE : TO COMPUTE DERIVATIVES AND POTENTIAL FOR TRAJECTORY MODULE.
C*********************************************************************
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      INTEGER PERSEC,SECIND,PERPAN,PANIND
     1         PSAVE(144,10),CSAVE(144,10),PSECTR(144),CSECTR(144),
      COMMON/BLK1/
     1DELEE(20),DELMAX(144),EMAX(144),ETA(20),BSCAT1(144),
     2BSCAT2(144),BSCAT3(144),PHO(20,40),
     1ANG(40),CPHOMX(144),SUNANG,ANGO,BOUNDL(20,40),BOUNDI(20,40),
     2ANGL(20,40),VII(20,40),ICHECK(20,40),RATIO(13),IROOT(144),
     3IFAULT(14),JLOCAL(144),ICON(144),
     1VI(20),EV(8),NLEV,NTIMES,ILEV,TIMIN(20),TIMAX(20),VIMUM(20),
     2A(144),B(144),C(144),BB(144),BW(144),TW(144),DW(144),
     3EW(144),TWOCOS(700),
     1IFLG,NPOINT,M,IGEOM,ISYM,IPOIS,NSECT,MPEROD,MPEROD,IDIMY,
     2GAMMA,DVDX(70,48),DVDY(70,48),Y(144),DELTAX,DELTAY,
     3YS(66,48),XS(66),XSS(70),
     1VARBLE(4),T,DELT,SBOUND,NDIM,POT,N,J1,RPI,RPI2,RPIHAF,SQTPI,
     2SAY,TEMPE1,TEMPE2,DENE1,DENE2,DENI1,DENI2,TEMPI1,TEMPI2,
     3TEMPR,LUN3,LUN4,LUN5,LUN6,LUN7,LUN8,LUN9,LUN10,
     1IDTL(144),JDTL(144),PERSEC(144),SECIND(144),IPANEL(144),
     2PERPAN(144,4),PANIND(144),
     1CION(144),CELEC(144),CSEC(144),CSEC1(144),CPHOTO(144),
     2CPHOT1(144),CBSCAT(144),CBSCA1(144),SECPRN(144),SECPR1(144),
     3PHI(144),COUN(144),
     1KASE,JOML,DPHI,NANO,RAD,PBOUND,OMEGA,ALPHA,SO,JP,IS,ECS,
     2EXPVI2(20),EFTVI(20),IERROR,TETMIN,TETMAX,BETA1,BETA2,JLESS,
     3L2,CARRIV,CPH,XI,CSECEL,CSECE1,CS,CB,CS1,CB1
      IBETA1=0
      DO 1 K=1,NDIM
    1 V(K)=VARBLE(K)
   10 I=IFIX(VARBLE(1)/DELTAX)+1
      ORB=VARBLE(3)
   11 IF(ORB.LT.0.0) ORB=ORB+RPI2
      IF(ORB.GE.RPI2) ORB=ORB-RPI2
      IF(ORB.LT.0.0.OR.ORB.GE.RPI2) GO TO 11
      DO 12 J=1,N
      IF(ORB.LT.Y(1)) GO TO 13
      IF(ORB.GE.Y(N)) GO TO 14
      IF(ORB.GE.Y(J).AND.ORB.LT.Y(J+1)) GO TO 15
   12 CONTINUE
      GO TO 15
   13 A=ORB+RPI2
      J=N
      JP1=1
      GO TO 16
   14 J=N
      JP1=1
      A=ORB
      GO TO 16
C
   15 JP1=J+1
      IF(J.EQ.N) JP1=1
      A=ORB
   16 CONTINUE
      S=VARBLE(1)
      IP1=I+1
C DO BILINEAR INTERPOLATION ON POTENTIAL DERIVATIVES
      AA=(S-XS(I))/DELTAX
      BB=(A-Y(J))/DELTAY
      MP2=M+2
      DO 31 II=1,MP2
      IF(VARBLE(1).GE.XSS(II).AND.VARBLE(1).LT.XSS(II+1)) GO TO 32
   31 CONTINUE
   32 III=II+1
      AAA=(S-XSS(II))/(XSS(II+1)-XSS(II))
      DVXA=(1.0-AAA)*(1.0-BB)*DVDX(II,J)+BB*(1.0-AAA)*DVDX(II,JP1)+
     1 AAA*(1.0-BB)*DVDX(III,J)+AAA*BB*DVDX(III,JP1)
C
      DO 33 JJ=1,N
      IF (ORB.LT.(Y(1)+0.5*DELTAY)) GO TO 34
      IF (ORB.GE.(Y(N)+0.5*DELTAY)) GO TO 35
      JJ1=JJ+1
      IF(JJ.EQ.N) JJ1=1
      IF(ORB.GE.(Y(JJ)+0.5*DELTAY).AND.ORB.LT.
     1 Y(JJ1)+0.5*DELTAY) GO TO 36
   33 CONTINUE
C
   34 JJ=N
      JJ1=1
      AO=ORB+RPI2
      GO TO  37
C
   35 JJ=N
      JJ1=1
      AO=ORB
      GO TO 37
C
   36 AO=ORB
   37 BBB=(AO-Y(JJ)-0.5*DELTAY)/DELTAY
      DVYA=(1.0-AA)*(1.0-BBB)*DVDY(I,JJ)+BBB*(1.0-AA)*DVDY(I,JJ1)+
     1 AA*(1.0-BBB)*DVDY(IP1,JJ)+AA*BBB*DVDY(IP1,JJ1)
C DO BILINEAR INTERPOLATION ON POTENTIAL
      AA=(EXP(S)-EXP(XS(I)))/(EXP(XS(I+1))-EXP(XS(I)))
      POT=(1.0-AA)*(1.0-BB)*YS(I,J)+BB*(1.0-AA)*YS(I,JP1)+
     1 AA*(1.0-BB)*YS(IP1,J)+AA*BB*YS(IP1,JP1)
C
      EXPU=EXP(-VARBLE(1))
   30 BETA2=BETA1-POT/GAMMA
      IF(BETA2.LE.0.0) IBETA1=1
      IF(BETA2.LE.0.) RETURN
      BEX=0.5*EXP(-2.0*VARBLE(1))
      DERY(1)=VARBLE(2)
      DERY(4)=VARBLE(4)**2-VARBLE(2)**2-BEX*DVXA/GAMMA
      DERY(3)=-2.0*VARBLE(2)*VARBLE(4)-BEX*DVYA/GAMMA
      DERY(4)=VARBLE(4)
      RETURN
      END
      SUBROUTINE GUESS(PHI,PSECTR,CSECTR,Y)
C
C
C
C
C
C
C                        PACK# GE # 6
C                     --------------------
C
C         GUESS FIELD COMPUTATION
C
C
C
C
C
C
C
C
C*************************************************************************
C PURPOSE: TO COMPUTE A GUESS FIELD.
C*************************************************************************
C
      INTEGER PERSEC,SECIND,PERPAN,PANIND
      COMMON PSAVE(144,10),CSAVE(144,10),PSECTR(144),CSECTR(144),
     1DELEE(20),DELMAX(144),EMAX(144),ETA(20),BSCAT1(144),
     2BSCAT2(144),BSCAT3(144),PHO(20,40),
     1ANG(40),CPHOMX(144),SUNANG,ANGO,BOUNDL(20,40),BOUNDI(20,40),
     2ANGL(20,40),VII(20,40),ICHECK(20,40),RATIO(13),IROOT(144),
     3IFAULT(14),JLOCAL(144),ICON(144),
     1VI(20),EV(8),NLEV,NTIMES,ILEV,TIMIN(20),TIMAX(20),VIMUM(20),
     2A(144),B(144),C(144),BB(144),BW(144),TW(144),DW(144),
     3EW(144),TWOCOS(700),
     1IFLG,NPOINT,M,IGEOM,ISYM,IPOIS,NSECT,MPEROD,MPEROD,IDIMY,
     2GAMMA,DVDX(70,48),DVDY(70,48),Y(144),DELTAX,DELTAY,
     3YS(66,48),XS(66),XSS(70),
     1VARBLE(4),T,DELT,SBOUND,NDIM,POT,N,J1,RPI,RPI2,RPIHAF,SQTPI,
     2SAY,TEMPE1,TEMPE2,DENE1,DENE2,DENI1,DENI2,TEMPI1,TEMPI2,
     3TEMPR,LUN3,LUN4,LUN5,LUN6,LUN7,LUN8,LUN9,LUN10,
     1IDTL(144),JDTL(144),PERSEC(144),SECIND(144),IPANEL(144),
     2PERPAN(144,4),PANIND(144),
     1CION(144),CELEC(144),CSEC(144),CSEC1(144),CPHOTO(144),
     2CPHOT1(144),CBSCAT(144),CBSCA1(144),SECPRN(144),SECPR1(144),
     3PHI(144),COUN(144),
     1KASE,JOML,DPHI,NANO,RAD,PBOUND,OMEGA,ALPHA,SO,JP,IS,ECS,
```

```
     2EXPVI2(20),EFTVI(20),IERROR,TETMIN,TETMAX,BETA1,BETA2,JLESS,
     3L2,CARRIV,CPH,XI,CSECEL,CSECE1,CS,CB,CS1,CB1
      INTEGER I,J,K,L,M,N,T
      DATA TOL/5.0E-3/
      DATA ISWITH/0/
C
      DO 1 I=1,NSECT
      CURN(I)=0.0
      CURP(I)=0.0
      PHIN(I)=.0
      PHIP(I)=.0
    1 CONTINUE
C
      NTOUR=-1
   10 NTOUR=NTOUR+1
      IF (NTOUR.GE.800) WRITE (9,600)
  600 FORMAT (1X,*NO CONVERGENCE IN GUESS FIELD AFTER 800 ITERS*)
      IF (NTOUR.GE.800) STOP
      L=0
      DO 100 I=1,NSECT
      ECS=0.0
      CET(I)=0.0
      CN=0.0
      K=PERSEC(I)
      N=L+K
      T=L+1
C
      DO 90 J=T,N
      M=SECIND(J)
      ECS=ECS+CURENT(M,PHI(M),Y(M),CE,CPH,CS,CB,CSECEL)
      CET(I)=CET(I)+CE
      PSECTR(I)=PHI(M)
   90 CONTINUE
      L=L+K
      CSECTR(I)=ECS
  100 CONTINUE
C
C PREPARE DATA FOR REGULAR FALSI METHOID
      IF (NTOUR-1) 201,401,401
  201 DO 210 I=1,NSECT
      IF(CSECTR(I).GT.0.0) GO TO 111
      CURN(I)=CSECTR(I)
      PHIN(I)=PSECTR(I)
      PSECTR(I)=PSECTR(I)-5.0
      GO TO 210
  111 CURP(I)= CSECTR(I)
      PHIP(I)= PSECTR(I)
      PSECTR(I)=PSECTR(I)+1.0
  210 CONTINUE
C
      GO TO 500
C
C BEGIN REGULA FALSI ITERATION
  401 K=0
      DO 410 I=1,NSECT
      IF(ABS(CSECTR(I)).LE.TOL*CET(I)) GO TO 410
      IF(CURN(I).EQ.0.0) PHIN(I)=PSECTR(I)
      IF (CURN(I).EQ.0.0) CURN(I)=CSECTR(I)
      IF (CURP(I).EQ.0.0) PHIP(I)=PSECTR(I)
      IF (CURP(I).EQ.0.0) CURP(I)=CSECTR(I)
      K=K+1
      IF (CURN(I)*CSECTR(I).GE.0.0) GO TO 420
      CURP(I)=CSECTR(I)
      PHIP(I)=PSECTR(I)
      GO TO430
  420 CURN(I)=CSECTR(I)
      PHIN(I)=PSECTR(I)
  430 IF(CURN(I)*CURP(I).LE.0.0) GO TO 440
      IF(CURN(I).GE.0.0) PSECTR(I)=PSECTR(I)+1.0
      IF (CURP(I).LE.0.0) PSECTR(I)=PSECTR(I)-1.0
      GO TO 410
  440 PSECTR(I)=(PHIN(I)*CURP(I)-PHIP(I)*CURN(I))/(CURP(I)-CURN(I
     1 ))
  410 CONTINUE
      IF (K.NE.0) GO TO 500
      WRITE (9,401)
  401 FORMAT (1X,* OK PAST GUESS FIELD*)
      WRITE (9,6001) (PHI(J),J=1,NPOINT)
 6001 FORMAT (6(1X,1PE12.5))
      ISWITH=1
C
  500 L=0
      DO 501 I=1,NSECT
      PHINEW=PSECTR(I)
      K=PERSEC(I)
      N=L+K
      T=L+1
      DO 510 J=T,N
      M=SECIND(J)
      PHI(M)=PHINEW
  510 CONTINUE
      L=L+K
  501 CONTINUE
      IF(ISWITH.EQ.1) RETURN
      GO TO 10
      END
      FUNCTION CURENT (J,XI,YY,CE,CPH,CS,CB,CSECEL)
C*************************************************************************
C PURPOSE: TO COMPUTE CURENT COLLECTION AT GRID POINTS FOR GUESS FIELD.
C*************************************************************************
      INTEGER PERSEC,SECIND,PERPAN,PANIND
      COMMON PSAVE(144,10),CSAVE(144,10),PSECTR(144),CSECTR(144),
     1DELEE(20),DELMAX(144),EMAX(144),ETA(20),BSCAT1(144),
     2BSCAT2(144),BSCAT3(144),PHO(20,40),
     1ANG(40),CPHOMX(144),SUNANG,ANGO,BOUNBL(20,40),BOUNBI(20,40),
     2ANGL(20,40),VII(20,40),ICHECK(20,40),RATIO(13),IROOT(144),
     3IFAULT(14),JLOCAL(144),ICON(144),
     1VI(20),EV(8),NLEV,NTIMES,ILEV,TIMIN(20),TIMAX(20),VIMUM(20),
     2A(144),B(144),C(144),BB(144),BW(144),TW(144),BM(144),
     3EW(144),TWOCOS(700),
     1IFLG,NPOINT,M,IGEOM,ISYM,IPOIS,NSECT,NPEROD,NPEROD,IDIMY,
     2GAMMA,DVDX(70,48),DVDY(70,48),Y(144),DELTAX,DELTAY,
     3YS(66,48),XS(66),XSS(70),
     1VARBLE(4),T,DELT,SBOUND,NDIM,POT,N,J1,RPI,RPI2,RPIHAF,SQTPI,
     2SAY,TEMPE1,TEMPE2,DENE1,DENE2,DENI1,DENI2,TEMPI1,TEMPI2,
     3TEMPR,LUN3,LUN4,LUN5,LUN6,LUN7,LUN8,LUN9,LUN10,
     1IDTL(144),JDTL(144),PERSEC(144),SECIND(144),IPANEL(144),
     2PERPAN(144,4),PANIND(144),
     1CION(144),CELEC(144),CSEC(144),CSEC1(144),CPHOTO(144),
     2CPHOT1(144),CBSCAT(144),CBSCA1(144),SECPRN(144),SECPR1(144),
     3PHI(144),COUN(144),
     1KASE,JOHL,DPHI,NANG,RAD,PBOUND,OMEGA,ALPHA,SO,JP,IS,ECS,
     2EXPVI2(20),EFTVI(20),IERROR,TETMIN,TETMAX,BETA1,BETA2,JLESS,
     3L2,CARRIV,CPH,XI,CSECEL,CSECE1,CS,CB,CS1,CB1
C
      DATA FAC/1.0/
      CSECEL=0.0
      CI1=0.0
      CI2=0.0
      CPH=0.0
      CE1=0.0
      CE2=0.0
      CS=0.0
      CB=0.0
      CS1=0.0
      CB1=0.0
      IF(IFAULT(3).NE.0.OR.IFAULT(4).NE.0) CALL CBS(XI,J,CS,CB,CS1,CB1)
      IF(IFAULT(9).EQ.1) CALL SECEL(XI,CSECEL,CSECE1)
      VAR4=YY-SUNANG
      IF (XI.GT.0.0) GO TO 1
      IF(RATIO(2).NE.0.0) CE1=RATIO(6)*EXP(-XI/RATIO(2))
      IF(RATIO(11).NE.0.0) CE2=RATIO(11)*EXP(-XI/RATIO(10))
      CE=(CE1*DENE1+CE2*DENE2)/DENI1
      CI1=2.0*SAY*(SQRT(-XI/RATIO(8))+COEFT(SQRT(-XI/RATIO(8))))
      CI1=CI1*RATIO(7)
      IF (TEMPI2.NE.0.0)
     1 CI2=(2.0*SAY*SQRT(-XI/RATIO(1))+COEFT(SQRT(-XI/RATIO(1))))*
     2 RATIO(12)*DENI2/DENI1
      CI=CI1+CI2
      IF(IFAULT(5).EQ.1)
     1CPH=0.5*CPHOMX(J)*(COS(VAR4)+ABS(COS(VAR4)))*FAC*RATIO(9)
      CURENT=CI+CPH-CE+CS+CB+CSECEL
      IF(ITRACE.EQ.0)
     1WRITE (9,600) J,XI,CE1,CE2,CI,CPH,CS,CB,CSECEL,CURENT
      RETURN
C
    1 CI1=EXP(-XI/RATIO(8))*RATIO(7)
      IF(TEMPI2.NE.0.0) CI2=EXP(-XI/RATIO(1))*RATIO(12)*DENI2/DENI1
      CI=CI1+CI2
      IF(RATIO(2).NE.0.0)
     1CE1=RATIO(6)*(2.0*SAY*(SQRT(-XI/RATIO(2))+COEFT(SQRT(-XI/RATIO(2)
     1 ))))
      IF(RATIO(10).NE.0.0)
     1CE2=RATIO(11)*(2.*SAY*(SQRT(-XI/RATIO(10))+COEFT(SQRT(-XI/
     2 RATIO(10) ))))
      CE=(CE1*DENE1+CE2*DENE2)/DENI1
      IF(IFAULT(5).EQ.1)
     1CPH=0.5*CPHOMX(J)*(COS(VAR4)+ABS(COS(VAR4)))*EXP(XI/RATIO(5))
      CPH=CPH*2.0*SAY*(SQRT(-XI/RATIO(5))+COEFT(SQRT(-XI/
     1 RATIO(5))))*FAC*RATIO(9)
      CURENT=CI+CPH-CE+CS+CB+CSECEL
      IF(ITRACE.EQ.0)
     1WRITE (9,600) J,XI,CE1,CE2,CI,CPH,CS,CB,CSECEL,CURENT
  600 FORMAT (1X,I6,9(1PE10.3))
      RETURN
      END
      SUBROUTINE SECTOR(NPOINT,NSECT,PERSEC,SECIND,PHI,PSECTR,
     1 CSECTR,CION,CELEC,CSEC,CBSCAT,CPHOTO,SECPRN,ILOOP,ITER,DEBACC)
C
C*************************************************************
C OUTPUT
C       PSECTR,CSECTR
C DESCRIPTION OF VARIABLES
C       NPOINT          NUMBER OF POINTS
C       NSECT           NUMBER OF SECTORS
C       PERSEC(I)       NUMBER OF POINTS IN SECTOR I,1<I<NSECT
C       SECIND          VECTOR OF INDICES  DIMENSION NPOINT
C       INDICES ARE A PERMUTATION OF INTEGERS J,1=<J=<NPOINT
C        FIRST PERSEC(1) ELEMENTS ARE INDICES OF POINTS IN SECTOR 1
C       NEXT PERSEC (2) ELEMENTS INDICES OF POINT IN SECTOR 2,ETC
C*********************************************************
C
      INTEGER T
      INTEGER PERSEC,SECIND,PERPAN,PANIND
      COMMON PSAVE(144,10),CSAVE(144,10),PSECTR(144),CSECTR(144),
     1DELEE(20),DELMAX(144),EMAX(144),ETA(20),BSCAT1(144),
     2BSCAT2(144),BSCAT3(144),PHO(20,40),
     1ANG(40),CPHOMX(144),SUNANG,ANGO,BOUNBL(20,40),BOUNBI(20,40),
     2ANGL(20,40),VII(20,40),ICHECK(20,40),RATIO(13),IROOT(144),
     3IFAULT(14),JLOCAL(144),ICON(144),
     1VI(20),EV(8),NLEV,NTIMES,ILEV,TIMIN(20),TIMAX(20),VIMUM(20),
     2A(144),B(144),C(144),BB(144),BW(144),TW(144),BM(144),
     3EW(144),TWOCOS(700),
     1IFLG,NPOINT,M,IGEOM,ISYM,IPOIS,NSECT,NPEROD,NPEROD,IDIMY,
     2GAMMA,DVDX(70,48),DVDY(70,48),Y(144),DELTAX,DELTAY,
     3YS(66,48),XS(66),XSS(70),
     1VARBLE(4),T,DELT,SBOUND,NDIM,POT,N,J1,RPI,RPI2,RPIHAF,SQTPI,
     2SAY,TEMPE1,TEMPE2,DENE1,DENE2,DENI1,DENI2,TEMPI1,TEMPI2,
     3TEMPR,LUN3,LUN4,LUN5,LUN6,LUN7,LUN8,LUN9,LUN10,
     1IDTL(144),JDTL(144),PERSEC(144),SECIND(144),IPANEL(144),
     2PERPAN(144,4),PANIND(144),
     1CION(144),CELEC(144),CSEC(144),CSEC1(144),CPHOTO(144),
     2CPHOT1(144),CBSCAT(144),CBSCA1(144),SECPRN(144),SECPR1(144),
     3PHI(144),COUN(144),
     1KASE,JOHL,DPHI,NANG,RAD,PBOUND,OMEGA,ALPHA,SO,JP,IS,ECS,
     2EXPVI2(20),EFTVI(20),IERROR,TETMIN,TETMAX,BETA1,BETA2,JLESS,
     3L2,CARRIV,CPH,XI,CSECEL,CSECE1,CS,CB,CS1,CB1
      L=0
      DO 100 I=1,NSECT
      ECS=0.0
      K=PERSEC(I)
      N=L+K
      T=L+1
      DO 90 J=T,N
      MM=SECIND(J)
      ECS=ECS+CETT(MM)
      PSECTR(I)=PHI(MM)
   90 CONTINUE
      L=L+K
      CSECTR(I)=ECS
  100 CONTINUE
      WRITE (8,601) (CSECTR(I),I=1,NSECT)
  601 FORMAT (8(1X,1PE10.3))
      RETURN
      END
      FUNCTION CETT(J)
C***************************************************************
C ALL CURRENTS ARE DIMENSIONALIZED WITH RESPECT TO REF TEMP
C        CION            N-D ION CURRENT
C        CELEC           N-D ELECTRON CURRENT
```

```
C       CSEC            N-D SEC CURRENT
C       BSCAT BACK-SCATTER ELECTRON CURRENT
C       CPHOTO          N-D PHOTO ELECTRON CURRENT
C       CETT           NET CURRENT
C       SECPRN  N-D SEC ELECTRON CURRENT INDUCED BY PROTONS
C       XI              N-D POTENTIAL
C*********************************************************************
      INTEGER PERSEC,SECIND,PERPAN,PANIND
      COMMON PSAVE(144,10),CSAVE(144,10),PSECTR(144),CSECTR(144),
     1DELEE(20),DELMAX(144),EMAX(144),ETA(20),BSCAT1(144),
     2BSCAT2(144),BSCAT3(144),PHO(20,40),
     1ANG(40),CPHOMX(144),SUNANG,ANGO,BOUNDL(20,40),BOUNDI(20,40),
     2ANGL(20,40),VII(20,40),ICHECK(20,40),RATIO(13),IROOT(144),
     3IFAULT(14),JLOCAL(144),ICON(144),
     1VI(20),EV(8),NLEV,NTIMES,ILEV,TIMIN(20),TIMAX(20),VINUM(20),
     2A(144),B(144),C(144),BB(144),BW(144),TW(144),DW(144),
     3EW(144),TWOCOS(700),
     1IFLG,NPOINT,M,IGEOM,ISYM,IPOIS,NSECT,NPEROD,MPEROD,IDIMY,
     2GAMMA,DVDX(70,48),DVDY(70,48),Y(144),DELTAX,DELTAY,
     3YS(66,48),XS(66),XSS(70),
     1VARBLE(4),T,DELT,SBOUND,NDIM,POT,H,J1,RPI,RPI2,RPIHAF,SQTPI,
     2SAY,TEMPE1,TEMPE2,DENE1,DENE2,DENI1,DENI2,TEMPI1,TEMPI2,
     3TEMPR,LUN3,LUN4,LUN5,LUN6,LUN7,LUN8,LUN9,LUN10,
     1IDTL(144),JDTL(144),PERSEC(144),SECIND(144),IPANEL(144),
     2PERPAN(144,4),PANIND(144),
     1CION(144),CELEC(144),CSEC(144),CSEC1(144),CPHOTO(144),
     2CPHOT1(144),CBSCAT(144),CBSCA1(144),SECPRN(144),SECPR1(144),
     3PHI(144),CGUN(144),
     1KASE,JOML,DPHI,NANG,RAD,PBOUND,OMEGA,ALPHA,SO,JP,IS,ECS,
     2EXPVI2(20),EFTVI(20),IERROR,TETMIN,TETMAX,BETA1,BETA2,JLESS,
     3L2,CARRIV,CPH,XI,CSECEL,CSECE1,CS,CB,CS1,CB1
C
      IF (IFAULT(1).EQ.0) CION(J)=0.0
      IF (IFAULT(2).EQ.0) CELEC(J)=0.0
      IF (IFAULT(3).EQ.0) CSEC(J)=0.0
      IF (IFAULT(4).EQ.0) CBSCAT(J)=0.0
      IF (IFAULT(5).EQ.0) CPHOTO(J)=0.0
      CPHOTO(J)=CPHOTO(J)*RATIO(9)
      CPHOT1(J)=CPHOT1(J)*RATIO(9)
      IF( IFAULT(9).EQ.0) SECPRN(J)=0.0
      CETT=CION(J)-CELEC(J)+CSEC(J)+CBSCAT(J)+CPHOTO(J)+SECPRN(J)
      WRITE (LUN4,600) J,PHI(J),CETT,CION(J),CELEC(J),CPHOTO(J),CPHOT1(J
     1 ),CSEC(J),CSEC1(J),CBSCAT(J),CBSCA1(J),SECPRN(J),SECPR1(J)
  600 FORMAT(I3,12(1PE10.3))
      RETURN
      END
      SUBROUTINE CBS(XI,J,CS,CB,CS1,CB1)
C*********************************************************************
C PURPOSE IS TO FIND THE CONTRIBUTION OF SEC AND BACKS ELECTRONS TO THE
C GUESS FIELD.
C
C*********************************************************************
      INTEGER PERSEC,SECIND,PERPAN,PANIND
      COMMON PSAVE(144,10),CSAVE(144,10),PSECTR(144),CSECTR(144),
     1DELEE(20),DELMAX(144),EMAX(144),ETA(20),BSCAT1(144),
     2BSCAT2(144),BSCAT3(144),PHO(20,40),
     1ANG(40),CPHOMX(144),SUNANG,ANGO,BOUNDL(20,40),BOUNDI(20,40),
     2ANGL(20,40),VII(20,40),ICHECK(20,40),RATIO(13),IROOT(144),
     3IFAULT(14),JLOCAL(144),ICON(144),
     1VI(20),EV(8),NLEV,NTIMES,ILEV,TIMIN(20),TIMAX(20),VINUM(20),
     2A(144),B(144),C(144),BB(144),BW(144),TW(144),DW(144),
     3EW(144),TWOCOS(700),
     1IFLG,NPOINT,M,IGEOM,ISYM,IPOIS,NSECT,NPEROD,MPEROD,IDIMY,
     2GAMMA,DVDX(70,48),DVDY(70,48),Y(144),DELTAX,DELTAY,
     3YS(66,48),XS(66),XSS(70),
     1VARBLE(4),T,DELT,SBOUND,NDIM,POT,H,J1,RPI,RPI2,RPIHAF,SQTPI,
     2SAY,TEMPE1,TEMPE2,DENE1,DENE2,DENI1,DENI2,TEMPI1,TEMPI2,
     3TEMPR,LUN3,LUN4,LUN5,LUN6,LUN7,LUN8,LUN9,LUN10,
     1IDTL(144),JDTL(144),PERSEC(144),SECIND(144),IPANEL(144),
     2PERPAN(144,4),PANIND(144),
     1CION(144),CELEC(144),CSEC(144),CSEC1(144),CPHOTO(144),
     2CPHOT1(144),CBSCAT(144),CBSCA1(144),SECPRN(144),SECPR1(144),
     3PHI(144),CGUN(144),
     1KASE,JOML,DPHI,NANG,RAD,PBOUND,OMEGA,ALPHA,SO,JP,IS,ECS,
     2EXPVI2(20),EFTVI(20),IERROR,TETMIN,TETMAX,BETA1,BETA2,JLESS,
     3L2,CARRIV,CPH,XI,CSECEL,CSECE1,CS,CB,CS1,CB1
C
      DO 710 JJ=1,2
      IF(JJ.NE.1) GO TO 711
      GAMMA=RATIO(2)
      TEMPE=TEMPE1
      RATE=RATIO(6)
      DEN=DENE1/DENI1
      GO TO 712
  711 GAMMA=RATIO(10)
      TEMPE=TEMPE2
      RATE=RATIO(11)
      DEN=DENE2/DENI1
  712 CONTINUE
      IF(DEN.EQ.0.0) GO TO 710
      NUM=1
      VZERO=0.0
      IF (XI.NE.0.0)
     1VZERO=AMAX1(0.,-XI/GAMMA*SQRT(ABS(XI/GAMMA))/ABS(XI/GAMMA
     1 ))
      DO 705 I=1,NLEV,ILEV
      BETA1=VI(I)**2
      BETA2=BETA1-XI/GAMMA
      IF (BETA2.LE.0.0) GO TO 705
      NUM=NUM+1
      VINUM(NUM)=SQRT(BETA2)
      DELEE(NUM)=7.4*DELMAX(J)*BETA2*TEMPE*EXP(-2.0*SQRT(BETA2*TEMPE/
     1 EMAX(J)))/EMAX(J)
      BSC=BSCAT3(J)*BETA2*GAMMA
      ETA(NUM)=BSCAT1(J)
      IF (BSC.LE.100.0)
     1 ETA(NUM)=BSCAT1(J)-BSCAT2(J)*EXP(-BSC)
  705 CONTINUE
      IF (NUM.EQ.1) GO TO 710
      VINUM(1)=VZERO
      BETA2=VZERO**2
      DELEE(1)=7.4*DELMAX(J)*BETA2*TEMPE*EXP(-2.0*SQRT(BETA2*TEMPE/
     1 EMAX(J)))/EMAX(J)
      BSC=BSCAT3(J)*BETA2*GAMMA
      ETA(1)=BSCAT1(J)
      IF (BSC.LE.100.0)
     1 ETA(1)=BSCAT1(J)-BSCAT2(J)*EXP(-BSC)
      NUMM1=NUM-1
      CUR2=0.0
      CUR3=0.0
      COEF1=EXP(-XI/GAMMA)/SQTPI
      ADELO=0.0
      BDELO=0.0
      ASCAO=0.0
      BSCAO=0.0
C
      DO 816 I=1,NUMM1
      DENOM=VINUM(I+1)-VINUM(I)
      ADELI=(DELEE(I+1)-DELEE(I))/DENOM
      BDELI=DELEE(I)-ADELI*VINUM(I)
      ADEL1=2.0*ADELI
      BDEL1=2.0*BDELI
      CUR2=CUR2+EXP(-VINUM(I)**2)*((ADEL1-ADELO)*(1.0+VINUM(I)**2)
     1 + (BDEL1-BDELO)*(VINUM(I)+COEFT(VINUM(I))))
      ASCAI=(ETA(I+1)-ETA(I))/DENOM
      BSCAI=ETA(I)-ASCAI*VINUM(I)
      ASCA1=2.0*ASCAI
      BSCA1=2.0*BSCAI
      CUR3=CUR3+EXP(-VINUM(I)**2)*((ASCA1-ASCAO)*(1.0+VINUM(I)**2)+
     1 (BSCA1-BSCAO)*(VINUM(I)+COEFT(VINUM(I))))
      ADELO=ADEL1
      BDELO=BDEL1
      ASCAO=ASCA1
      BSCAO=BSCA1
  816 CONTINUE
C
      CUR22=CUR2
      CUR33=CUR3
      IF (XI.LE.0.0) GO TO 817
      CUR2=(2.0*SAY*(SQRT(-XI/RATIO(3))+COEFT(SQRT(-XI/RATIO(3)))))*
     1 CUR2*EXP(XI/RATIO(3))
      CUR3=(2.0*SAY*(SQRT(-XI/RATIO(4))+COEFT(SQRT(-XI/RATIO(4)))))*
     1 CUR3*EXP(XI/RATIO(4))
C
  817 CONTINUE
      IF(IFAULT(3).EQ.0.0) CUR2=0.0
      IF (IFAULT(4).EQ.0.0) CUR3=0.0
      CS=RATE*COEF1*CUR2*DEN+CS
      CB=RATE*COEF1*CUR3*DEN+CB
      CS1=CS1+RATE*COEF1*CUR22*DEN
      CB1=CB1+RATE*COEF1*CUR33*DEN
  710 CONTINUE
      RETURN
      END
      SUBROUTINE SECEL(XI,CSECEL,CSECE1)
C
C*********************************************************************
CPURPOSE
C       TO COMPUTE SECONDARY ELECTRONS INDUCED BY PROTONS FOR THE
C       GUESS FIELD
C
C
C EXTERNALS NONE
C*********************************************************************
      INTEGER PERSEC,SECIND,PERPAN,PANIND
      COMMON PSAVE(144,10),CSAVE(144,10),PSECTR(144),CSECTR(144),
     1DELEE(20),DELMAX(144),EMAX(144),ETA(20),BSCAT1(144),
     2BSCAT2(144),BSCAT3(144),PHO(20,40),
     1ANG(40),CPHOMX(144),SUNANG,ANGO,BOUNDL(20,40),BOUNDI(20,40),
     2ANGL(20,40),VII(20,40),ICHECK(20,40),RATIO(13),IROOT(144),
     3IFAULT(14),JLOCAL(144),ICON(144),
     1VI(20),EV(8),NLEV,NTIMES,ILEV,TIMIN(20),TIMAX(20),VINUM(20),
     2A(144),B(144),C(144),BB(144),BW(144),TW(144),DW(144),
     3EW(144),TWOCOS(700),
     1IFLG,NPOINT,M,IGEOM,ISYM,IPOIS,NSECT,NPEROD,MPEROD,IDIMY,
     2GAMMA,DVDX(70,48),DVDY(70,48),Y(144),DELTAX,DELTAY,
     3YS(66,48),XS(66),XSS(70),
     1VARBLE(4),T,DELT,SBOUND,NDIM,POT,H,J1,RPI,RPI2,RPIHAF,SQTPI,
     2SAY,TEMPE1,TEMPE2,DENE1,DENE2,DENI1,DENI2,TEMPI1,TEMPI2,
     3TEMPR,LUN3,LUN4,LUN5,LUN6,LUN7,LUN8,LUN9,LUN10,
     1IDTL(144),JDTL(144),PERSEC(144),SECIND(144),IPANEL(144),
     2PERPAN(144,4),PANIND(144),
     1CION(144),CELEC(144),CSEC(144),CSEC1(144),CPHOTO(144),
     2CPHOT1(144),CBSCAT(144),CBSCA1(144),SECPRN(144),SECPR1(144),
     3PHI(144),CGUN(144),
     1KASE,JOML,DPHI,NANG,RAD,PBOUND,OMEGA,ALPHA,SO,JP,IS,ECS,
     2EXPVI2(20),EFTVI(20),IERROR,TETMIN,TETMAX,BETA1,BETA2,JLESS,
     3L2,CARRIV,CPH,XI,CSECEL,CSECE1,CS,CB,CS1,CB1
      CSECE1=0.0
      CSECEL=0.0
      DO 710 JJ=1,2
      IF(JJ.NE.1) GO TO 711
      GAMMA=RATIO(8)
       TEMPI=TEMPI1
      DEN=+1.0
      RATE=RATIO(7)
      GO TO 712
  711 CONTINUE
      GAMMA=RATIO(1)
      TEMPI=TEMPI2
      DEN=DENI2/DENI1
      RATE=RATIO(12)
  712 CONTINUE
      IF(TEMPI.EQ.0.0) GO TO 710
      NUM=1
      VZERO=0.0
      IF(XI.NE.0.0)
     1 VZERO=AMAX1(0.0,-XI/GAMMA*SQRT(ABS(XI/GAMMA))/
     2 ABS(XI/GAMMA))
C
      DO 700 I=1,NLEV,ILEV
      BETA1=VI(I)**2
      BETA2=BETA1-XI/GAMMA
      IF(BETA2.LE.0.0) GO TO 700
      NUM=NUM+1
      VINUM(NUM)=SQRT(BETA2)
      DELEE(NUM)=DELP(J)*SQRT(0.001*BETA2*TEMPI)/(1.0+BETA2*TEMPI/
     1 EXMP(J))
  700 CONTINUE
C
      IF(NUM.EQ.1) GO TO 710
      VINUM(1)=VZERO
      BETA2=VINUM(1)**2
      DELEE(1)=DELP(J)*SQRT(0.001*BETA2*TEMPI)/(1.0+BETA2*TEMPI/
     1 EXMP(J))
C
      CUR2=0.0
      COFF1=EXP(-XI/GAMMA)/SQTPI
      ADELO=.0
      BDELO=0.0
      NUMM1=NUM-1
      DO 816 I=1,NUMM1
      DENOM=VINUM(I+1)-VINUM(I)
      ADELI=(DELEE(I+1)-DELEE(I))/DENOM
      BDELI=DELEE(I)-ADELI*VINUM(I)
      ADEL1=2.0*ADELI
      BDEL1=2.0*BDELI
      CUR2=CUR2+EXP(-VINUM(I)**2)*((ADEL1-ADELO)*(1. )+VINUM(I)**2)
     1 +  (BDEL1-BDELO)*(VINUM(I)+COEFT(VINUM(I))))
      ADELO=ADEL1
      BDELO=BDEL1
  816 CONTINUE
      CUR22=CUR2
      IF (XI.GT.0.0) CUR2=(2.0*SAY*(SQRT(-XI/RATIO(13))+
     1 COEFT(SQRT(-XI/RATIO(13)))))*EXP(XI/RATIO(13))*CUR2
      CSECEL=CSECEL+COFF1*CUR2*RATE*DEN
      CSECE1=CSECE1+COFF1*CUR22*RATE*DEN
  710 CONTINUE
      RETURN
      END
      SUBROUTINE POI(IFLG,NPEROD,N,MPEROD,M,A,B,C,IDIMY,Y,IERROR,W)
C
C
C
```

Appendix D: Listing of plotting program used with CYLVIA

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C           PROGRAM CYPLOT
C
C  THIS PROGRAM SOLVES POISSON S EQUATION OUTSIDE A CIRCLE(SPACECRAFT)
C  ON WHICH POTENTIAL IS GIVEN VS ANGLE, AND GENERATES A CONTOUR MAP
C  OF THE RESULT.
C
C  TEMPR=REFERENCE TEMPERATURE(VOLTS).
C  RAD=SPACECRAFT RADIUS (METERS).
C  IGEOM=1  MEANS GEOMETRY IS INFINITE-CYLINDRICAL.
C  IGEOM=2  MEANS GEOMETRY IS PROLATE SPHEROIDAL.
C  RPLOT IS PLOTTED SPACECRAFT RADIUS(INCHES).
C  SIGMA=MAJOR-TO-MINOR-AXIS RATIO OF PROLATE SPHEROID.
C  SBOUND=NATURAL LOG OF (OUTER GRID BOUNDARY RADIUS/SPACECRAFT RADIUS),
C    IF IGEOM=1.
C  PBOUND=POTENTIAL AT OUTER BOUNDARY(IN KV).
C  CNT IS THE SAME AS THE PARAMETER 'SEP' USED BY SUBROUTINE 'CNTOUR'.
C    ITS VALUE DETERMINES WHETHER AND HOW CONTOURS ARE LABELLED.
C  DANGO IS THE ANGLE(IN DEGREES) BETWEEN THE ORIGIN OF ANGLES AND J=1.
C  M IS NUMBER OF UNKNOWNS(INTERIOR GRID NODES) IN I-DIRECTION(RADIAL
C    DIRECTION), I.E., IT IS THE NUMBER OF NODES NOT INCLUDING THE
C    SPACECRAFT SURFACE OR THE OUTER GRID BOUNDARY.
C  N IS THE NUMBER OF UNKNOWNS(GRID POINTS) IN J-DIRECTION(IN ANGLE).
C  KSPOTS=0  MEANS SURFACE POTENTIAL VALUES ARE READ IN.
C    KSPOTS=1 MEANS HALF OF THEM ARE READ IN AND SYMMETRY IS USED
C    TO GENERATE THE OTHER HALF.
C   KSPOTS AT LEAST 2 MEANS THEY ARE GENERATED BY SUBROUTINE 'SPOTS'.
C  MODCHG=0  MEANS ZERO SPACE CHARGE IS ASSUMED.MODCHG=1 MEANS LINEAR
C    SPACE CHARGE IS ASSUMED.
C  ISCALE=0 MEANS RADIAL SCALE IS LOGARITHMIC IN REAL RADIUS.
C  ISCALE=1 MEANS RADIAL SCALE IS PROPORTIONAL TO REAL RADIUS.
C  NCN IS THE NUMBER OF POTENTIAL VALUES FOR WHICH EQUIPOTENTIAL
C    CONTOURS ARE TO BE PLOTTED.
C  CN(1,2,....,NCN) ARE POTENTIAL VALUES (IN KV) FOR WHICH CONTOURS
C    TO BE PLOTTED.
C  TEMPE1,....,DENI2  ARE TEMPERATURES (IN VOLTS) AND DENSITIES (IN
C    PARTICLES PER CUBIC METER) FOR ELECTRONS AND IONS IN ASSUMED
C    DOUBLE-MAXWELLIAN VELOCITY DISTRIBUTIONS.
C  PHI(1,2,....,N) ARE SURFACE POTENTIAL VALUES(IN KV).
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      COMMON /CTRACE/RATIO(10),JROUTE(32),ITRACE,IFAULT(10),JLOCAL(10)
      COMMON/CPOIS/A(66),B(66),C(66),TWOCOS(1500),IFLG
      COMMON/POISX/NPEROD,MPEROD,IDIMY
      COMMON/CPOISA/N,M,KSPOTS,MODCHG
      COMMON/CDIV/YS(66,180),XS(70),SY(66,181)
      COMMON /CORB/VARBLE(4),T,DELT,SBOUND,FUNC(4),OKUTA(4),H
      COMMON/PIE/ RPI,RPI2,RPIHAF,SQTPI,SAY
      COMMON/POT/GAMMA,DELTAX,DELTAY,Y(180),RAD,RPLOT
      COMMON/BLK1/TEMPE1,TEMPE2,DENE1,DENE2,DENI1,DENI2,TEMPI1,TEMPI2
     1,TEMPR
      COMMON/BLK2/PHI(180),DELPHI(180),CN(20),NCN,ISCALE
C
      READ(5,40) TEMPR,RAD
   40 FORMAT(2F10.1)
      READ(5,41) IGEOM,RPLOT,SIGMA
   41 FORMAT(I10,2F10.1)
      READ(5,40) SBOUND,PBOUND
      READ(5,42) CNT,DANGO,M,N,KSPOTS,MODCHG,ISCALE
   42 FORMAT(2F6.2,5I4)
C
      WRITE(6,50) TEMPR,RAD,IGEOM,RPLOT,SIGMA,SBOUND,PBOUND,CNT,DANGO,
     1M,N,KSPOTS,MODCHG,ISCALE
   50 FORMAT(1X,'CYPLOT',//6X,'TEMPR',7X,'RAD',5X,'IGEOM',5X,'RPLOT',
     15X,'SIGMA',4X,'SBOUND',4X,'PBOUND',7X,'CNT',5X,'DANGO',
     27X,'M',7X,'N',2X,'KSPOTS',2X,'MODCHG',2X,'ISCALE'/
     31X,1P2E10.3,I10,1P6E10.3,5I8//)
      READ(5,400) NCN
  400 FORMAT(I2)
      IF(NCN.NE.0) READ(5,401) (CN(K),K=1,NCN)
  401 FORMAT(8F10.7)
      IF(NCN.NE.0) WRITE(6,402) (CN(K),K=1,NCN)
  402 FORMAT(1X,'POTENTIAL CONTOURS IN KV'/(1X,10F10.5/))
C
C NPEROD IS ALWAYS ZERO, GIVES PERIODIC BOUNDARY CONDITION IN
C   ANGLE FOR POISSON-SOLVER.
C
      NPEROD=0
      RPI=3.14159265335897
      RPI2=2.0*RPI
      RPIHAF=0.5*RPI
      SQTPI=1.772453851
      IPOIS=0
      IFLG=0
      SUNANG=0.0
      HCONS=10.0
      SAY=1.0/SQTPI
      READ(5,500) TEMPE1,TEMPE2,DENE1,DENE2
  500 FORMAT(4E11.3)
      WRITE(6,600) TEMPE1,TEMPE2,DENE1,DENE2
  600 FORMAT(8X,6HTEMPE1,6X,6HTEMPE2,6X,5HDENE1,7X,5HDENE2/5X,1P4E12.3/)
      READ(5,500) TEMPI1,TEMPI2,DENI1,DENI2
      WRITE(6,601) TEMPI1,TEMPI2,DENI1,DENI2
  601 FORMAT(8X,6HTEMPI1,6X,6HTEMPI2,6X,5HDENI1,7X,5HDENI2/5X,1P4E12.3/)
      ANGO=DANGO*RPI/180.0
C
C *** IF THE FIRST DIMENSION OF THE ARRAY YS IS CHANGED, IDIMY MUST BE
C     CHANGED. ***
C
      IDIMY=66
      MP2=M+2
      IF(KSPOTS.EQ.0) READ(5,200) (PHI(J),J=1,N)
  200 FORMAT(6E12.5)
      IF(KSPOTS.NE.1) GO TO 202
      NHALF=N/2
      READ(5,200) (PHI(J),J=1,NHALF)
      DO 203 J=1,NHALF
      PHI(N+1-J)=PHI(J)
  203 CONTINUE
      GO TO 204
  202 IF(KSPOTS.GE.2) CALL SPOTS
  204 WRITE(6,603)
  603 FORMAT(//1X,'***SURFACE POTENTIALS IN KV***')
      WRITE (6,201) (PHI(J),J=1,N)
  201 FORMAT(1X,1P10E12.5)
      IF(IGEOM.EQ.1)
     1CALL GENGRD(TEMPI,TEMPE,DENSTE,SBOUND,PBOUND
     1,M,HCONS,ANGO,IPOIS)
      IF(IGEOM.EQ.2)
     1CALL PROLAT(TEMPI,SBOUND,PBOUND,M,
     2 HCONS,ANGO,IPOIS,SIGMA,RMAX,OMEGA,ALPHA,SMAX,SO)
      CALL BARIER(MP2,N)
      IF(NCN.EQ.0) STOP
      CALL PLOPOT(SBOUND,PBOUND,M,N,SUNANG,CNT)
      CALL SADDLE
      STOP
      END
      SUBROUTINE GENGRD(TEMPI,TEMPE,DENSTE,SBOUND,PBOUND
     1,M,HCONS,ANGO,IPOIS)
      COMMON/CPOIS/A(66),B(66),C(66),TWOCOS(1500),IFLG
      COMMON/POISX/NPEROD,MPEROD,IDIMY
      COMMON/CPOISA/N,M,KSPOTS,MODCHG
      COMMON/CDIV/YS(66,180),XS(70),SY(66,181)
      COMMON/PIE/ RPI,RPI2,RPIHAF,SQTPI,SAY
      COMMON/POT/GAMMA,DELTAX,DELTAY,Y(180),RAD,RPLOT
      COMMON/BLK1/TEMPE1,TEMPE2,DENE1,DENE2,DENI1,DENI2,TEMPI1,TEMPI2
     1,TEMPR
      COMMON/BLK2/PHI(180),DELPHI(180),CN(20),NCN,ISCALE
      IF(IFLG.EQ.1)GO TO 100
      RLMDA=0.
      IF(MODCHG.EQ.0) GO TO 212
      DEBYE=69.*SQRT(TEMPR*1.602E-19/(DENI1*1.381E-23))
      RLMDA=(RAD/DEBYE)**2*(DENE1/TEMPE1+DENE2/TEMPE2+DENI1/TEMPI1+
     1 DENI2/TEMPI2)        *TEMPR/DENI1
      RATIO=1.0/SQRT(RLMDA)
      WRITE(6,211) RATIO
  211 FORMAT(/1X,'DEBYE LENGTH / SPACECRAFT RADIUS = '1PE10.3//)
  212 DELTAX=SBOUND/ FLOAT(M+1)
      DELTAY=2.*RPI/ FLOAT(N)
      SYX=(DELTAY/DELTAX)**2
C
C GENERATE GRID POINTS, COEFFICIENTS AND RHS OF EQUATION (I.NE.1)
C
  100 MM1=M-1
      DO 200 J=1,N
      Y(J)=FLOAT(J-1)*DELTAY+ANGO
      IF(Y(J).GT.RPI2) Y(J)=Y(J)-RPI2
      YS(M,J)=-SYX*PBOUND
      DO 210 I=2,MM1
      YS(I,J)=0.
  210 CONTINUE
  200 CONTINUE
      TLMDA=RLMDA*DELTAY**2
      DO 220 I=1,M
      XS(I)= FLOAT(I)*DELTAX
      A(I)=SYX
      B(I)=-2.*SYX-TLMDA* EXP(2.*XS(I))
      C(I)=SYX
  220 CONTINUE
      A(1)=0.
      C(M)=0.
C
C CALCULATION OF RHS OF EQUATION FOR I=1
C
  300 DO 230 J=1,N
      YS(1,J)=-SYX*PHI(J)
  230 CONTINUE
      CALL POIS(IFLG,NPEROD,N,MPEROD,M,A,B,C,IDIMY,YS,IERROR,TWOCOS)
      WRITE(6,1001) IERROR
 1001 FORMAT(1X,'THE ERROR FLAG IS ',I3)
      WRITE(6,1002)
 1002 FORMAT(1X,'***POTENTIAL PROFILE***'/)
      DO 130 J=1,N
      WRITE(6,120) J,(YS(I,J),I=1,M)
  120 FORMAT(1X,2HJ=,I3,/(1X,1P13E10.3))
  130 CONTINUE
      RETURN
      END
      SUBROUTINE PROLAT  (TEMPI,SBOUND,PBOUND,
     1 M,HCONS,ANGO,IPOIS,SIGMA,RMAX,OMEGA,ALPHA,SMAX,SO)
C
C**********************************************************************
C PURPOSE
C      TO GENERATE ARRAY OF GRID POINTS AND TO COMPUTE THE COEFFICIENTS
C      OF THE POISSON EQUATION IN THE CASE OF A PROLATE SPHEROID.
C
C DESCRIPTION OF THE VARIABLES
C      RMAX    IS OUTER RADIUS AT WHICH POTENTIAL IS ASSUMED TO BE ZERO.
C      SIGMA   IS POLAR AXIS LENGTH TO EQUATORIAL AXIS LENGTH.
C
C EXTERNALS
C      POI
C
C**********************************************************************
C
      COMMON/CPOIS/A(66),B(66),C(66),TWOCOS(1500),IFLG
      COMMON/POISX/NPEROD,MPEROD,IDIMY
      COMMON/CPOISA/N,M,KSPOTS,MODCHG
      COMMON/CDIV/YS(66,180),XS(70),SY(66,181)
      COMMON/PIE/ RPI,RPI2,RPIHAF,SQTPI,SAY
      COMMON/POT/GAMMA,DELTAX,DELTAY,Y(180),RAD,RPLOT
      COMMON/BLK1/TEMPE1TEMPE2,DENE1,DENE2,DENI1,DENI2,TEMPI1,TEMPI2
     1,TEMPR
      COMMON/BLK2/PHI(180),DELPHI(180),CN(20),NCN,ISCALE
C
      RLMDA=0.0
      DELTAY=RPI2/FLOAT(N)
      TAUP=0.5*ALOG((SIGMA+1.0)/(SIGMA-1.0))
      ALPHA=1.0/SINH(TAUP)
      OMEGA=COSH(TAUP)
      BETA=EXP(TAUP)
      SO=ALOG((BETA+1.0)/(BETA-1.0))
      SO=-SO
      THETA=RMAX/ALPHA
      TAUMAX=ALOG(THETA+SQRT(THETA**2+1.0))
      BETA=EXP(TAUMAX)
      SMAX=ALOG((BETA+1.0)/(BETA-1.0))
      SMAX=-SMAX
      SBOUND=SMAX
      DELTAX=(SMAX-SO)/FLOAT(M+1)
      SYX=(DELTAY/DELTAX)**2
      MM1=M-1
      DO 200 J=1,N
      Y(J)=FLOAT(J-1)*DELTAY+ANGO
      IF (Y(J).GE.RPI2) Y(J)=Y(J)-ANGO
      IF (Y(J).LT.0.0) Y(J)=Y(J)+RPI2
      YS(M,J)=-SYX*PBOUND
      DO 210 I=2,MM1
       YS(I,J)=0.0
  210 CONTINUE
  200 CONTINUE
      MP2=M+2
      DO 220 I=1,M
      XS(I)=FLOAT(I)*DELTAX+SO
      BETA=EXP(-XS(I))
      TAU=ALOG((BETA+1.0)/(BETA- 1.0))
      PSI=1.0/(COSH(TAU))**2
      A(I)=SYX*PSI
      B(I)=SYX*PSI
      C(I)=SYX*PSI
  220 CONTINUE
      A(1)=0.0
      C(M)=0.0
      DO 230 J=1,N
      YS(1,J)=-SYX*PHI(J)
  230 CONTINUE
      CALL POIS(IFLG,NPEROD,N,MPEROD,M,A,B,C,IDIMY,YS,IERROR,TWOCOS)
```

```
      WRITE (6,             )HA,DELTAY,SMAX,SO,OMEGA
  600 FORMAT (5(1X,1PE10.3))
      H=(DELTAX*DELTAY)/((DELTAX+DELTAY)*HCONS)
      RETURN
      END
      SUBROUTINE PLOPOT(SBOUND,PBOUND,M,NPOINT,SUNANG,CNT)
C
C POTENTIAL CONTOURS AROUND A SPACECRAFT UNDER MAGNETOSPHERIC
C  CONDITIONS.
C
      DIMENSION X(70)
      DIMENSION YY(181)
      COMMON/CDIV/YS(66,180),XS(70),SY(66,181)
      COMMON/PIE/RPI,RPI2,RPIHAF,SQTPI,SAY
      COMMON/POT/GAMMA,DELTAX,DELTAY,Y(180),RAD,RPLOT
      COMMON/BLK2/PHI(180),DELPHI(180),CN(20),NCN,ISCALE
      DATA FAC/1000./
      RP=RPLOT
      N1=NPOINT+1
      MP2=M+2
      MP1=M+1
      DELTAX=SBOUND/FLOAT(MP2-1)
C
C RELOCATE ELEMENTS OF THE ARRAYS XS AND YS.
C
      DO 300 J=1,NPOINT
      YS(MP2,J)=PBOUND*FAC
      DO 310 I=2,MP1
      IPA=MP2-I
      IPB=IPA+1
      YS(IPB,J)=YS(IPA,J)*FAC
  310 CONTINUE
      YS(1,J)=PHI(J)*FAC
  300 CONTINUE
      DO 320 I=1,MP2
  320 XS(I)=DELTAX*FLOAT(I-1)
      IF(DELTAX/DELTAY.LT.1.6) GO TO 99
C
C GRID CELLS ARE TOO ELONGATED RADIALLY. FIX THIS USING INTERPOLATION.
C
      MP1=MINO(MP1*2,64)
      MP2=MP1+1
      M=MP1-1
      MM1=M-1
      MM2=M-2
      DELTAX=DELTAX/2.0
      DO 91 I=1,MP2,2
      IV=MP2+1-I
      KV=(IV+1)/2
      DO 91 J=1,NPOINT
   91 YS(IV,J)=YS(KV,J)
      DO 92 J=1,NPOINT
      YS(2,J)=(3.0*YS(1,J)+6.0*YS(3,J)-YS(5,J))/8.0
      YS(MP1,J)=(3.0*YS(MP2,J)+6.0*YS(M,J)-YS(MM2,J))/8.0
      DO 92 I=4,MM1,2
   92 YS(I,J)=(9.0*(YS(I-1,J)+YS(I+1,J))-YS(I-3,J)-YS(I+3,J))/16.0
      DO 93 I=1,MP2
   93 XS(I)=DELTAX*FLOAT(I-1)
      WRITE(6,90) M
   90 FORMAT(//1X,'RADIAL GRID INTERVAL HALVED. NEW M =',I3)
C
C SET UP EXPANDED ARRAY OF POTENTIAL VALUES FOR USE BY PLOTTER.
C
   99 DO 1 I=1,MP2
      SY(I,N1)=YS(I,1)
      DO 1 J=1,NPOINT
      SY(I,J)=YS(I,J)
    1 CONTINUE
C
C GENERATE GRID
C
      DO 2 J=1,NPOINT
      YY(J)=Y(J)
    2 CONTINUE
      YY(N1)=RPI2+Y(1)
C
      DO 3 I=1,MP2
      IF(ISCALE.EQ.0) X(I)=RP*(1.0+FLOAT(I-1)*DELTAX)
      IF(ISCALE.EQ.1) X(I)=RP*EXP(FLOAT(I-1)*DELTAX)
    3 CONTINUE
C
C INITIALIZE PLOTTER
C
      CALL START
      CALL SYMBOL(0.0,10.5,0.15,'PLEASE MOUNT BLACK FINE POINT PEN IN 1'
     1,270.0,38)
      CALL PLOT(1.0,0.0,-3)
      CALL PLOT(8.0,5.5,-3)
C
C *** IF THE FIRST DIMENSION OF THE ARRAY SY IS CHANGED, IDX MUST BE
C     CHANGED. ***
C
      IDX=66
      DO 100 I=1,NCN
      CN(I)=CN(I)*FAC
      CALL CNTOUR (X,MP2,YY,N1,SY,IDX,CN(I),CNT,CN(I))
  100 CONTINUE
C
C
      AROHED=AMIN1(X(MP2)+DELTAX,6.5)
      SIZE=1.0
      CALL PLOT(AROHED,0.0,3)
      CALL PLOT(AROHED+0.375*SIZE,0.5*SIZE,2)
      CALL PLOT(AROHED+0.375*SIZE,0.25*SIZE,2)
      CALL PLOT(AROHED+0.75*SIZE,0.25*SIZE,2)
      CALL PLOT(AROHED+0.75*SIZE,-0.25*SIZE,2)
      CALL PLOT(AROHED+0.375*SIZE,-0.25*SIZE,2)
      CALL PLOT(AROHED+0.375*SIZE,-0.5*SIZE,2)
      CALL PLOT(AROHED,0.0,2)
C
C
      CALL SYMBOL(AROHED-0.5*SIZE,0.625*SIZE,0.25*SIZE,'SUNLIGHT',0.0,8)
      CALL SYMBOL(-0.75*SIZE,0.125*SIZE,0.25*SIZE,'SPACE-',0.0,6)
      CALL SYMBOL(-0.625*SIZE,-0.375*SIZE,0.25*SIZE,'CRAFT',0.0,5)
C
      CALL CIRCL(RP,0.0,0.0,360.0,RP,RP,0.0)
C
      DO 102 I=1,MP2
      DO 102 J=1,NPOINT
      YS(I,J)=YS(I,J)/FAC
  102 CONTINUE
      RETURN
      END
      SUBROUTINE TOURT(XR,YR,XS,YS)
      RADR=YR
      RADS=YS
      YR=XR*SIN(RADR)
      XR=XR*COS(RADR)
      YS=XS*SIN(RADS)
      XS=XS*COS(RADS)
      RETURN
      END
```

```
      SUBROUTINE BARIER(MP2,N)
C
C*******************************************************************
C PURPOSE
C       TO COMPUTE HEIGHT AND RADIAL DISTANCE OF POTENTIAL BARRIERS
C       BY FITTING A PARABOLA.
C DESCRIPTION OF THE VARIABLES
C       XBAR AND YBAR ARE BARRIER LOCATION AND HEIGHT
C EXTERNALS= NONE
C**********************************************************************
      COMMON/CDIV/YS(66,180),XS(70),SY(66,181)
      COMMON/PIE/ RPI,RPI2,RPIHAF,SQTPI,SAY
      COMMON/POT/GAMMA,DELTAX,DELTAY,Y(180),RAD,RPLOT
      DIMENSION BRADP(180),BRADN(180),BDEGP(180),BDEGN(180),BPOTP(180),
     1BPOTN(180),IP(180),IN(180),JP(180),JN(180)
      MP1=MP2-1
      NPB=0
      NNB=0
      DO 400 J=1,N
      IF (YS(1,J).GT.YS(MP2,J))GO TO 405
      YUMIN=YS(1,J)
      IRMIN=1
      YUMAX=YS(MP2,J)
      IRMAX=MP2
      GO TO 403
  405 YUMAX=YS(1,J)
      IRMAX=1
      YUMIN=YS(MP2,J)
      IRMIN=MP2
  403 DO 410 I=2,MP1
      IF(YS(I,J).GT.YUMAX) IRMAX=I
      IF(YS(I,J).GT.YUMAX) YUMAX=YS(I,J)
      IF(YS(I,J).LT.YUMIN) IRMIN=I
      IF(YS(I,J).LT.YUMIN)YUMIN=YS(I,J)
  410 CONTINUE
      IF(IRMAX.EQ.1.AND.IRMIN.EQ.MP2)GO TO 400
      IF(IRMAX.EQ.MP2.AND.IRMIN.EQ.1)GO TO 400
C
C A BARRIER HAS BEEN FOUND AT ANGLE J.
C
      DO 400 K=1,2
       GO TO(401,402),K
  401 IF (IRMAX.EQ.1.OR.IRMAX.EQ.MP2)GO TO 400
      I=IRMAX
      NPB=NPB+1
      GO TO 460
  402 IF(IRMIN.EQ.1.OR.IRMIN.EQ.MP2) GO TO 400
      I=IRMIN
      NNB=NNB+1
C
C CALCULATION OF HEIGHT AND RADIAL LOCATION OF POTENTIAL BARRIER BY
C FITTING A PARABOLA TO THREE POINTS
C
  460 DIMI=(YS(I,J)-YS(I-1,J))/DELTAX
      DIPI=(YS(I+1,J)-YS(I,J))/DELTAX
      APAR=(DIPI-DIMI)/(2.*DELTAX)
      BPAR=DIPI-APAR*(XS(I)+XS(I+1))
      CPAR=YS(I,J)-APAR*XS(I)**2-BPAR*XS(I)
      XB=-BPAR/(2.*APAR)
      YB=APAR*XB**2+BPAR*XB+CPAR
      GO TO (411,412),K
  411 IP(NPB)=I
      JP(NPB)=J
      BRADP(NPB)=EXP(XB)
      BDEGP(NPB)=Y(J)*180.0/RPI
      BPOTP(NPB)=YB
      GO TO 400
  412 IN(NNB)=I
      JN(NNB)=J
      BRADN(NNB)=EXP(XB)
      BDEGN(NNB)=Y(J)*180.0/RPI
      BPOTN(NNB)=YB
  400 CONTINUE
C
      IF(NPB.EQ.0) GO TO 421
      WRITE(6,422)
  422 FORMAT(//1X,'POSITIVE BARRIERS')
      WRITE(6,429) (IP(L),JP(L),BRADP(L),BDEGP(L),BPOTP(L),L=1,NPB)
  429 FORMAT(/1X,3(4X,'I',3X,'J',2X,'RAD/RS',1X,'DEGREES',1X,
     1'HEIGHT(KV)')/(1X,3(2I4,0P2F8.3,1PE12.5)))
  421 IF(NNB.EQ.0) GO TO 423
      WRITE(6,424)
  424 FORMAT(//1X,'NEGATIVE BARRIERS')
      WRITE(6,429) (IN(L),JN(L),BRADN(L),BDEGN(L),BPOTN(L),L=1,NNB)
C
  423 RETURN
      END
      SUBROUTINE SADDLE
C PURPOSE IS TO IDENTIFY SADDLE POINTS
C
      COMMON/CPOISA/N,M,KSPOTS,NODCHG
      COMMON/CDIV/YS(66,180),XS(70),SY(66,181)
      COMMON/PIE/ RPI,RPI2,RPIHAF,SQTPI,SAY
      COMMON/POT/GAMMA,DELTAX,DELTAY,Y(180),RAD,RPLOT
      COMMON/BLK2/PHI(180),DELPHI(180),CN(20),NCN,ISCALE
      DIMENSION SX(66,180),DVDR(180),DVDY(180)
C
C
      DO 100 J=1,N
      DVDR(J)=(-YS(3,J)+4.0*YS(2,J)-3.0*PHI(J))/(2.0*DELTAX*RAD)
      JP=J+1
      JM=J-1
      IF(JP.GT.N) JP=1
      IF(JM.EQ.0) JM=N
      DVDY(J)=(PHI(JP)-PHI(JM))/(2.0*DELTAY*RAD)
  100 CONTINUE
C
      WRITE(6,600)
  600 FORMAT(///1X,'RADIAL ELECTRIC FIELD AT THE SURFACE (KV/METER)')
      WRITE(6,700) (DVDR(J),J=1,N)
  700 FORMAT(10(1X,1PE12.5))
      WRITE(6,800)
  800 FORMAT(///1X,'TANGENTIAL ELECTRIC FIELD AT THE SURFACE',
     1' (KV/METER)')
      WRITE(6,700) (DVDY(J),J=1,N)
      WRITE(6,850)
  850 FORMAT(///)
C
C
      MP1=M+1
      MP2=M+2
      DO 1 J=1,N
      DO 1 I=2,MP2
      SX(I,J)=999.
    1 CONTINUE
C
C
      DO 2 J=1,N
      DO 2 I=2,MP1
    6 P=(YS(I+1,J)-YS(I-1,J))/(2.0*DELTAX)
      JP1=J+1
      IF(J.EQ.N) JP1=1
```

```
      JM1=J-1
      IF(J.EQ.1) JM1=N
      Q=(YS(I,JP1)-YS(I,JM1))/(2.0*DELTAY)
      PX=(YS(I+1,J)-2.0*YS(I,J)+YS(I-1,J))/(DELTAX*DELTAX)
      QY=(YS(I,JP1)-2.0*YS(I,J)+YS(I,JM1))/(DELTAY*DELTAY)
      PY=(YS(I+1,JP1)-YS(I-1,JP1)-YS(I+1,JM1)+YS(I-1,JM1))/
     1 (DELTAX*DELTAY*4.0)
     2 QX=PY
C
      DEN=PX*QY-QX*PY
      IF(DEN.NE.0.0) SX(I,J)=(Q*PY-P*QY)/DEN
      IF(DEN.NE.0.0) SY(I,J)=(P*QX-Q*PX)/DEN
      IF(ABS(SX(I,J)).LT.DELTAX.AND.ABS(SY(I,J)).LT.DELTAY)
     1 GO TO 2
      SX(I,J)=999.
C
    2 CONTINUE
C
C IF THERE IS NO SADDLE POINT,RETURN
C
      ICH=0
      DO 10 J=1,N
      DO 10 I=2,NP1
      IF(SX(I,J).EQ.999.0) ICH=ICH+1
   10 CONTINUE
      IF(ICH.NE.0) GO TO 11
      WRITE(6,900)
  900 FORMAT(1X,'NO SADDLE POINT FOUND')
      RETURN
C
C FIND ACCURATE LOCATION OF SADDLE POINTS
C
   11 DO 20 J=1,N
      JP1=J+1
      IF(J.EQ.N) JP1=1
      DO 20 I=1,NP1
C
C CHECK IF A SADDLE POINT HAS BEEN OBSERVED IN THE GRID CELL ABOVE
C   AND TO THE RIGHT OF THE POINT(I,J).
C
      AMOYX=0.0
      AMOYY=0.0
      AN=0.0
      JM1=J-1
      IF(I.EQ.1) GO TO 22
      IF(SX(I,J).EQ.999.0.OR.SX(I,J).LT.0.0.OR.SY(I,J).LT.0.0)
     1 GO TO 21
      AN=AN+1.0
      AMOYX=AMOYX+SX(I,J)
      AMOYY=AMOYY+SY(I,J)
   21 IF(SX(I,JP1).EQ.999.0.OR.SX(I,JP1).LT.0.0.OR.SY(I,JP1).GT.0.0)
     1 GO TO 22
      AN=AN+1.0
      AMOYX=AMOYX+SX(I,JP1)
      AMOYY=AMOYY+SY(I,JP1)+DELTAY
C
   22 IF(I.EQ.NP1) GO TO 24
      IF(SX(I+1,J).EQ.999.0.OR.SX(I+1,J).GT.0.0.OR.SY(I+1,J).LT.0.0)
     1 GO TO 23
      AN=AN+1.0
      AMOYX=AMOYX+SX(I+1,J)+DELTAX
      AMOYY=AMOYY+SY(I+1,J)
   23 IF(SX(I+1,JP1).EQ.999.0.OR.SX(I+1,JP1).GT.0.0.OR.SY(I+1,JP1).GT.
     A0.0) GO TO 24
      AN=AN+1.0
      AMOYX=AMOYX+SX(I+1,JP1)+DELTAX
      AMOYY=AMOYY+SY(I+1,JP1)+DELTAY
   24 CONTINUE
      IF(AN.EQ.0.0) GO TO 20
      AXS=AMOYX/AN
      AYS=AMOYY/AN
      AAXX=AXS+XS(I)
      AAYY=AYS+Y(J)
      XDS=AXS/DELTAX
      YDS=AYS/DELTAY
      SADPOT=(YS(I+1,J)*XDS+YS(I,J)*(1.0-XDS))*(1.0-YDS)
     1+(YS(I+1,JP1)*XDS+YS(I,JP1)*(1.0-XDS))*YDS
C
      SADRAD=EXP(AAXX)
      SADDEG=AAYY*180.0/RPI
      WRITE(6,601) I,J,SADRAD,SADDEG,SADPOT,AN
  601 FORMAT(1X,'SADDLE POINT DETECTED IN CELL',2I3,' RAD/RS= ',
     AF7.3,' DEGREES= ',F7.3,' POTENTIAL(KV)= ',1PE12.5,
     B' NUMBER OF DETECTIONS ',0PF4.1)
      IF(AN.LT.4.0) GO TO 20
      IF(ISCALE.EQ.0) SPRAD=RPLOTS(1.0+AAXX)
      IF(ISCALE.EQ.1) SPRAD=RPLOT*EXP(AAXX)
      SADX=SPRAD*COS(AAYY)
      SADY=SPRAD*SIN(AAYY)
      IF(ABS(SADY).GT.5.25) GO TO 20
      IF(ABS(SADX).GT.8.3) GO TO 20
      CALL SYMBOL(SADX,SADY,0.28,4,45.0,0)
   20 CONTINUE
      CALL FINISH
      RETURN
      END
      SUBROUTINE SPOTS
C
C THIS SUBROUTINE SETS THE SURFACE POTENTIAL VALUES PHI(1,2,.....,N)
C  IN A MANNER DEPENDING ON THE VALUE OF KSPOTS.
C
      COMMON/CPOISA/N,M,KSPOTS,NOBCHG
      COMMON/BLK2/PHI(180),DELPHI(180),CH(20),NCH,ISCALE
C
      IF(KSPOTS.GT.2) GO TO 900
C
      IF(N.NE.180) GO TO 900
      DO 11 J=1,N
      PHI(J)=-0.07
      IF(J.GE.1.AND.J.LE.8) PHI(J)=-0.08
      IF(J.GE.42.AND.J.LE.53) PHI(J)=-0.06
      IF(J.GE.91.AND.J.LE.98) PHI(J)=-0.05
      IF(J.GE.128.AND.J.LE.139) PHI(J)=-0.05
   11 CONTINUE
      RETURN
C
  900 WRITE(6,901) N,KSPOTS
  901 FORMAT(1X,'SPOTS CALLED INCORRECTLY. N = ',I3,
     A' KSPOTS = 'I3)
      RETURN
      END
      END$
```

Appendix E: Listing of XYCIC

```
FTN4
$EMA(AEMA1,0)
      PROGRAM OBJCT(3,150)
C
C  THIS IS THE MAIN ROUTINE FOR THE OBJECT DEFINITION SECTION
C   OF  XYCIC .
C
      COMMON/AEMA1/XG(2560),YG(2560),IG(2560,4),IL(53,53),
     1ICH(53,53),ICK(52,52)
      COMMON/CMESH/IMSHIN,JMSHIN,IMSHMX,JMSHMX,IMO,JMO,DX,DY
      COMMON/CNST1/NNEST(10),NNI(10),NNJ(10),NOUTER(10),KS(10),
     1KE(10),KAM(10),KAMA(10),MBW(10),MBWA(10)
      COMMON/CNST2/NESMAX,MAXNOD,MAXELE,NODOUT,NODIN,KELOUT,
     1KELIN,NODINS
      COMMON/CONN/NCONN(10,4,4)
      COMMON/CREF/LMRN,LMRB(32),LMRX(32),LMRY(32)
      COMMON/CSUN/SANGL,NSUNB(32),XSUNB(32),YSUNB(32),ASUNS(64),
     1NSUNP(128),NSUN
      COMMON/IODAT/LUN,LU,ITRACS
      COMMON/OBJ1/NOBJCT,NSEGO(5),ISPACE(64),JSPACE(64),XSSEG(64),
     1YSSEG(64),XESEG(64),YESEG(64),ISLP(64),IVEC(64),SLPSEG(64),
     2CGOX(5),CGOY(5)
      COMMON/OBJ2/XSP(128),YSP(128),XEP(128),YEP(128),NSP(128),
     1NEP(128),NMP(32),NPAT(5),MAXPAT
      COMMON/PSOLV/ISYM,IPOISS,IBPOT,RBEDY,CPOIS
      COMMON/RPI/HPI,PI,HPI3,PI2,SQPI,SQPI2
      COMMON/BLK1/XS(15,6),YS(15,6),XE(15,6),YE(15,6),TY(15,7),
     1YYM1(15,5),YYM2(15,5)
      COMMON/BLK2/BUF1(15),BUF2(15),BUF3(15),BUF4(15)
      COMMON/BLK3/BUF5(15),BUF6(15),BUF7(15),BUF8(20),BUF9(30)
      COMMON/BLK4/IBUF1(25),IBUF2(25),IBUF3(25),IBUF4(25)
      COMMON/BLK5/LR(4,4),LT(5,4),KX(9),KY(9),KN(4),NCB(4,6),
     1LCL(15,9)
      COMMON/BLK6/XO,YO,NST,NS,DM1,DM2,DM3,DM4,DM5
      COMMON/BLK7/NO,IO,JO,NST,KO,LO,NC1,NC2,NC3,INN,JNN
      COMMON/BLK8/NS,NE,IDUM1,IDUM2,IDUM3,IDUM4,IDUM5,IDUM6
      COMMON/BLK9/II1,II2,II3,II4,II5,II6,II7,II8,II9,II10
      DIMENSION NAME(3)
      DATA NAME/2HID,2HAT,2H1 /
C
      WRITE(LUN,999)
  999 FORMAT(////10X,'***XYCIC* SATELLITE SIMULATION***')
      CALL EXEC(8,NAME)
C
      STOP
      END
      BLOCK DATA
      COMMON/CMESH/IMSHIN,JMSHIN,IMSHMX,JMSHMX,IMO,JMO,DX,DY
      COMMON/CNST1/NNEST(10),NNI(10),NNJ(10),NOUTER(10),KS(10),
     1KE(10),KAM(10),KAMA(10),MBW(10),MBWA(10)
      COMMON/CNST2/NESMAX,MAXNOD,MAXELE,NODOUT,NODIN,KELOUT,
     1KELIN,NODINS
      COMMON/CONN/NCONN(10,4,4)
      COMMON/CREF/LMRN,LMRB(32),LMRX(32),LMRY(32)
      COMMON/CSUN/SANGL,NSUNB(32),XSUNB(32),YSUNB(32),ASUNS(64),
     1NSUNP(128),NSUN
      COMMON/IODAT/LUN,LU,ITRACS
      COMMON/OBJ1/NOBJCT,NSEGO(5),ISPACE(64),JSPACE(64),XSSEG(64),
     1YSSEG(64),XESEG(64),YESEG(64),ISLP(64),IVEC(64),SLPSEG(64),
     2CGOX(5),CGOY(5)
      COMMON/OBJ2/XSP(128),YSP(128),XEP(128),YEP(128),NSP(128),
     1NEP(128),NMP(32),NPAT(5),MAXPAT
      COMMON/PSOLV/ISYM,IPOISS,IBPOT,RBEDY,CPOIS
      COMMON/RPI/HPI,PI,HPI3,PI2,SQPI,SQPI2
      COMMON/BLK1/XS(15,6),YS(15,6),XE(15,6),YE(15,6),TY(15,7),
     1YYM1(15,5),YYM2(15,5)
      COMMON/BLK2/BUF1(15),BUF2(15),BUF3(15),BUF4(15)
      COMMON/BLK3/BUF5(15),BUF6(15),BUF7(15),BUF8(20),BUF9(30)
      COMMON/BLK4/IBUF1(25),IBUF2(25),IBUF3(25),IBUF4(25)
      COMMON/BLK5/LR(4,4),LT(5,4),KX(9),KY(9),KN(4),NCB(4,6),
     1LCL(15,9)
      COMMON/BLK6/XO,YO,NST,NS,DM1,DM2,DM3,DM4,DM5
      COMMON/BLK7/NO,IO,JO,NST,KO,LO,NC1,NC2,NC3,INN,JNN
      COMMON/BLK8/NS,NE,IDUM1,IDUM2,IDUM3,IDUM4,IDUM5,IDUM6
      COMMON/BLK9/II1,II2,II3,II4,II5,II6,II7,II8,II9,II10
C
C***LU# AND TRACK# WHERE I/O DATA STORAGE STARTS.
      DATA LUN/6/,LU/51/,ITRACS/0/
C***CONSTANTS WITH RESPECT TO 'PI'.
      DATA HPI,PI,HPI3,PI2,SQPI,SQPI2/1.57079632679497,
     A3.14159265358979,4.71238898038469,6.28318530717958,
     B1.77245385090551,1.12837916709551/
C
      END
      SUBROUTINE SORT(N)
      COMMON/BLK4/IA(25),IB(25),IBUF3(25),IBUF4(25)
C
      N1=N-1
      M=1
    1 M1=M+1
    2 IF(IA(M).LE.IA(M1)) GO TO 3
      I1=IA(M)
      I2=IB(M)
      IA(M)=IA(M1)
      IB(M)=IB(M1)
      IA(M1)=I1
      IB(M1)=I2
    3 M1=M1+1
      IF(M1.LE.N) GO TO 2
      M=M+1
      IF(M.LE.N1) GO TO 1
      RETURN
      END
      END$
```

```
FTN4
$EMA(AEMA1,0)
      PROGRAM OBJDF(5,150)
C
C  THIS S/P GENERATES OBJECTS IN AN INNERMOST GRID SPACE.
C
      COMMON/AEMA1/XG(2560),YG(2560),IG(2560,4),IL(53,53),
     1ICH(53,53),ICK(52,52)
      COMMON/CMESH/IMSHIN,JMSHIN,IMSHMX,JMSHMX,IMO,JMO,DX,DY
      COMMON/CNST1/NNEST(10),NNI(10),NNJ(10),NOUTER(10),KS(10),
     1KE(10),KAM(10),KAMA(10),MBW(10),MBWA(10)
      COMMON/CNST2/NESMAX,MAXNOD,MAXELE,NODOUT,NODIN,KELOUT,
     1KELIN,NODINS
      COMMON/IODAT/LUN,LU,ITRACS
      COMMON/OBJ1/NOBJCT,NSEGO(5),ISPACE(64),JSPACE(64),XSSEG(64),
     1YSSEG(64),XESEG(64),YESEG(64),ISLP(64),IVEC(64),SLPSEG(64),
     2CGOX(5),CGOY(5)
      COMMON/RPI/HPI,PI,HPI3,PI2,SQPI,SQPI2
      COMMON/BLK8/NS,NE,ICOUNT,IDUM2,IDUM3,IDUM4,IDUM5,IDUM6
      DIMENSION BUF(32),IBUF(64),NAME(3)
      DATA NAME/2HSH,2HAP,2HE /
C
      WRITE(LUN,999)
  999 FORMAT(//,20('*'),5X,'*OBJDF* STARTS',5X,20('*'))
      IMSHMX=IMSHIN+1
      JMSHMX=JMSHIN+1
      IMO=IMSHIN/2+1
      JMO=JMSHIN/2+1
      M1=0
      M2=0
      NNI(1)=IMSHIN
      NNJ(1)=JMSHIN
      IF(NESMAX.EQ.1) GO TO 60
      DO 50 I=2,NESMAX
      M=NNEST(I)*2
      NNI(I)=NNI(I-1)/2+M
      NNJ(I)=NNJ(I-1)/2+M
      M1=M1+NNI(I)*NNJ(I)-NNI(I-1)*NNJ(I-1)/4
      M2=M2+(NNI(I)+1)*(NNJ(I)+1)-(NNI(I-1)/2+1)*(NNJ(I-1)/2+1)
   50 CONTINUE
   60 KELOUT=M1
      KELIN=M1+1
      NODOUT=M2
      NODIN=M2+1
      ICOUNT=0
      WRITE(LUN,750) KELOUT,KELIN,NODOUT,NODIN
  750 FORMAT('**  KELOUT=',I4,3X,'KELIN=',I4,3X,'NODOUT=',I4,3X,
     1'NODIN=',I4)
C
      DO 200 I=1,NESMAX
      IBUF(I)=NNI(I)
      IBUF(I+10)=NNJ(I)
  200 CONTINUE
      CALL EXEC(2,LU,IBUF,64,ITRACS+2,50)
      IBUF(1)=IMSHMX
      IBUF(2)=JMSHMX
      IBUF(3)=IMO
      IBUF(4)=JMO
      IBUF(5)=NODOUT
      IBUF(6)=NODIN
      IBUF(7)=KELOUT
      IBUF(8)=KELIN
      CALL EXEC(2,LU,IBUF,64,ITRACS+2,55)
C
      DO 80 I=1,2560
      XG(I)=0.0
      YG(I)=0.0
   80 CONTINUE
      DO 90 I=1,2560
      DO 90 J=1,4
      IG(I,J)=0
   90 CONTINUE
      DO 100 I=1,IMSHMX
      DO 100 J=1,JMSHMX
      ICH(I,J)=0
  100 CONTINUE
      DO 110 I=1,IMSHIN
      DO 110 J=1,JMSHIN
      ICK(I,J)=0
  110 CONTINUE
C
      DO 140 I=1,NOBJCT
      WRITE(LUN,748) I
  748 FORMAT(5X,10('<'),3X,'OBJECT # =',I2)
      NE=NSEGO(I)
      NS=1
      IF(I.NE.1) NS=NSEGO(I-1)+1
      CX=0.0
      CY=0.0
C
      DO 150 N=NS,NE
      N1=N+1
      IF(N.EQ.NE) N1=NS
      ND=ISPACE(N1)-ISPACE(N)
      NC=JSPACE(N1)-JSPACE(N)
      IF(ND.EQ.0) GO TO 160
      ISLP(N)=NC*2/ND
      GO TO 170
  160 ISLP(N)=3
      ND=NC
  170 IF(ND.LT.0) IVEC(N)=-1
      IF(ND.GT.0) IVEC(N)=1
      XSSEG(N)=FLOAT(ISPACE(N))*DX
      YSSEG(N)=FLOAT(JSPACE(N))*DY
      XESEG(N)=FLOAT(ISPACE(N1))*DX
      YESEG(N)=FLOAT(JSPACE(N1))*DY
      II=ISLP(N)
      IF(II.EQ.3) GO TO 180
      A=ATAN(FLOAT(IABS(II))/2.0)
      GO TO 182
  180 A=HPI
  182 IF(II.EQ.0) GO TO 184
      A1=ISIGN(1,II)
      GO TO 190
  184 A1=1.0
  190 SL=FLOAT(1-IVEC(N))*HPI+A1*A
      IF(SL.LT.0.0) SL=SL+PI2
      SLPSEG(N)=SL
      CX=CX+XSSEG(N)
      CY=CY+YSSEG(N)
  150 CONTINUE
      A=FLOAT(NE-NS+1)
      CGOX(I)=CX/A
      CGOY(I)=CY/A
      WRITE(LUN,752) (ISLP(N),N=NS,NE)
  752 FORMAT('**  ISLP'/(20I5))
      WRITE(LUN,754) (IVEC(N),N=NS,NE)
  754 FORMAT('**  IVEC'/(20I5))
      WRITE(LUN,756) CGOX(I),CGOY(I)
  756 FORMAT('**  CENTER OF GRAVITY=',2F10.3)
      WRITE(LUN,758) (XSSEG(N),N=NS,NE)
  758 FORMAT('**  XSSEG'/(1P12E10.2))
      WRITE(LUN,760) (YSSEG(N),N=NS,NE)
  760 FORMAT('**  YSSEG'/(1P12E10.2))
      CALL UNIT
  140 CONTINUE
C
      CALL EXEC(2,LU,ISLP,64,ITRACS+2,56)
      CALL EXEC(2,LU,IVEC,64,ITRACS+2,58)
      CALL EXEC(2,LU,XSSEG,128,ITRACS+2,60)
      CALL EXEC(2,LU,YSSEG,128,ITRACS+2,64)
      CALL EXEC(2,LU,XESEG,128,ITRACS+2,68)
      CALL EXEC(2,LU,YESEG,128,ITRACS+2,72)
      CALL EXEC(2,LU,SLPSEG,128,ITRACS+2,76)
      DO 250 I=1,5
      BUF(I)=CGOX(I)
      BUF(I+10)=CGOY(I)
  250 CONTINUE
      CALL EXEC(2,LU,BUF,64,ITRACS+2,80)
      CALL EXEC(8,NAME)
C
      STOP
      END
$EMA(AEMA1,0)
```

```
      SUBROUTINE UNIT
      COMMON/AEMA1/XG(2560),YG(2560),IG(2560,4),IL(53,53),
     1ICH(53,53),ICK(52,52)
      COMMON/CMESH/INSHIN,JNSHIN,INSHMX,JNSHMX,INO,JNO,DX,DY
      COMMON/OBJ1/NOBJCT,NSEGO(5),ISPACE(64),JSPACE(64),XSSEG(64),
     1YSSEG(64),XESEG(64),YESEG(64),ISLP(64),IVEC(64),SLPSEG(64),
     2CGOX(5),CGOY(5)
      COMMON/BLK8/NS,NE,IDUM1,IDUM2,IDUM3,IDUM4,IDUM5,IDUM6
C
      DO 100 N=NS,NE
      IS=ISLP(N)
      IV=IVEC(N)
      II=IABS(IS)+1
      NXO=ISPACE(N)+INO
      NYO=JSPACE(N)+JNO
      IF(II.EQ.4) GO TO 110
      N1=NXO
      M=N+1
      IF(N.EQ.NE) M=NS
      N2=ISPACE(M)+INO
      GO TO 120
  110 N1=NYO
      M=N+1
      IF(N.EQ.NE) M=NS
      N2=JSPACE(M)+JNO
  120 M=1
      IF(II.EQ.2) M=2
      MM=IABS(N1-N2)/M
      ID=1
      IF(II.EQ.2) ID=2
C
      DO 130 M=1,MM
      IF(II.EQ.4) GO TO 140
      IO=NXO+ID*IV*(M-1)
      JO=NYO+IS*(IO-NXO)/2
      GO TO 150
  140 IO=NXO
      JO=NYO+IV*(M-1)
  150 ICH(IO,JO)=-3
      GO TO (160,180,220,240,250),II
C
  160 I1=IO+IV
      J1=JO+IV
      IF(ICH(IO,J1).GT.-2) ICH(IO,J1)=-1
      IF(ICH(I1,J1).GT.-2) ICH(I1,J1)=-1
      GO TO 130
C
  180 IF(IS.LT.0) GO TO 200
      I1=IO+IV
      I2=I1+IV
      J1=JO+IV
      IF(ICH(IO,J1).GT.-2) ICH(IO,J1)=-1
      IF(ICH(I1,J1).GT.-2) ICH(I1,J1)=-1
      IF(ICH(I1,JO).NE.-3) ICH(I1,JO)=-2
      IF(ICH(I2,JO).NE.-3) ICH(I2,JO)=-2
      IF(IV.EQ.-1) GO TO 190
      ICK(IO,JO)=5
      ICK(I1,JO)=5
      GO TO 130
  190 ICK(I1,J1)=7
      ICK(I2,J1)=7
      GO TO 130
C
  200 I1=IO+IV
      I2=I1+IV
      J1=JO-IV
      IF(ICH(I1,JO).GT.-2) ICH(I1,JO)=-1
      IF(ICH(I2,JO).GT.-2) ICH(I2,JO)=-1
      IF(ICH(I1,J1).NE.-3) ICH(I1,J1)=-2
      IF(ICH(IO,J1).NE.-3) ICH(IO,J1)=-2
      IF(IV.EQ.-1) GO TO 210
      ICK(IO,J1)=6
      ICK(I1,J1)=6
      GO TO 130
  210 ICK(I2,JO)=8
      ICK(I1,JO)=8
      GO TO 130
C
  220 IF(IS.LT.0) GO TO 230
      I1=IO+IV
      J1=JO+IV
      IF(ICH(IO,J1).GT.-2) ICH(IO,J1)=-1
      IF(ICH(I1,JO).NE.-3) ICH(I1,JO)=-2
      IF(IV.EQ.1) ICK(IO,JO)=1
      IF(IV.EQ.-1) ICK(I1,J1)=3
      GO TO 130
  230 I1=IO+IV
      J1=JO-IV
      IF(ICH(I1,JO).GT.-2) ICH(I1,JO)=-1
      IF(ICH(IO,J1).NE.-3) ICH(IO,J1)=-2
      IF(IV.EQ.1) ICK(IO,J1)=2
      IF(IV.EQ.-1) ICK(I1,JO)=4
      GO TO 130
C
  240 I1=IO-IV
      J1=JO+IV
      IF(ICH(I1,JO).GT.-2) ICH(I1,JO)=-1
      IF(ICH(I1,J1).GT.-2) ICH(I1,J1)=-1
      GO TO 130
C
  250 IF(IS.LT.0) GO TO 270
      I1=IO+IV
      J1=JO+IV
      J2=J1+IV
      IF(ICH(IO,J1).GT.-2) ICH(IO,J1)=-1
      IF(ICH(IO,J2).GT.-2) ICH(IO,J2)=-1
      IF(ICH(I1,J1).NE.-3) ICH(I1,J1)=-2
      IF(ICH(I1,JO).NE.-3) ICH(I1,JO)=-2
      IF(IV.EQ.-1) GO TO 260
      ICK(IO,JO)=9
      ICK(IO,J1)=9
      GO TO 130
  260 ICK(I1,J1)=11
      ICK(I1,J2)=11
      GO TO 130
C
  270 I1=IO+IV
      J1=JO-IV
      J2=J1-IV
      IF(ICH(I1,JO).GT.-2) ICH(I1,JO)=-1
      IF(ICH(I1,J1).GT.-2) ICH(I1,J1)=-1
      IF(ICH(IO,J1).NE.-3) ICH(IO,J1)=-2
      IF(ICH(IO,J2).NE.-3) ICH(IO,J2)=-2
      IF(IV.EQ.-1) GO TO 280
      J2=J1-IV
      ICK(IO,J1)=10
      ICK(IO,J2)=10
      GO TO 130
  280 ICK(I1,JO)=12
      ICK(I1,J1)=12
  130 CONTINUE
  100 CONTINUE
C
      RETURN
      END
      END$
```

```
FTN4
$EMA(AEMA1,0)
      PROGRAM SHAPE(5,150)
C
C THIS S/P GENERATES A MESH OF RECTANGULAR OR TRIANGULAR
C  FINITE ELEMENTS IN AN INNERMOST NEST AND DEFINE THE
C  NUMBERS OF THE NODES OF EACH ELEMENT. SURFACE NODES IS
C  ALSO DETERMINED.
C
      COMMON/AEMA1/XG(2560),YG(2560),IG(2560,4),IL(53,53),
     1ICH(53,53),ICK(52,52)
      COMMON/CMESH/INSHIN,JNSHIN,INSHMX,JNSHMX,INO,JNO,DX,DY
      COMMON/CNST1/NNEST(10),NNI(10),NNJ(10),NOUTER(10),KS(10),
     1KE(10),KAM(10),KAMA(10),NBW(10),NBWA(10)
      COMMON/CNST2/NESMAX,MAXNOD,MAXELE,NODOUT,NODIN,KELOUT,
     1KELIN,NODINS
      COMMON/CREF/LMRN,LMRD(32),LMRX(32),LMRY(32)
      COMMON/IODAT/LUN,LU,ITRACS
      COMMON/OBJ1/NOBJCT,NSEGO(5),ISPACE(64),JSPACE(64),XSSEG(64),
     1YSSEG(64),XESEG(64),YESEG(64),ISLP(64),IVEC(64),SLPSEG(64),
     2CGOX(5),CGOY(5)
      COMMON/BLK4/IBI(25),IBJ(25),IBUF3(25),IBUF4(25)
      DIMENSION IBUF(64),NAME(3)
      DATA NAME/2HLM,2HNE,2HF /
C
      WRITE(LUN,999)
  999 FORMAT(//20'.',5X,'"SHAPE" STARTS',5X,20'.')
      DO 20 I=1,INSHMX
      DO 30 J=1,JNSHMX
      J1=J+1
      IF(ICH(I,J).NE.-1) GO TO 30
      IF(ICH(I,J1).EQ.0) ICH(I,J1)=-1
   30 CONTINUE
   20 CONTINUE
C
      DO 40 I=1,INSHIN
      DO 50 J=1,JNSHIN
      IF(ICK(I,J).GE.1) GO TO 50
      IF(ICH(I,J).EQ.0.OR.ICH(I+1,J).EQ.0.OR.ICH(I+1,J+1).EQ.0.
     1OR.ICH(I,J+1).EQ.0) GO TO 50
      ICK(I,J)=-1
   50 CONTINUE
   40 CONTINUE
C
      MM=1
      IF(MM.EQ.1) GO TO 60
      DO 1300 J=1,JNSHMX
      JJ=JNSHMX+1-J
      WRITE(LUN,1350) (ICH(I,JJ),I=1,INSHMX)
 1300 CONTINUE
      DO 1310 J=1,JNSHIN
      JJ=JNSHIN+1-J
      WRITE(LUN,1350) (ICK(I,JJ),I=1,INSHIN)
 1310 CONTINUE
 1350 FORMAT(41I3)
C
   60 MI=0
      MJ=0
      NB1=NODOUT
      NINMAX=NNEST(1)
      IF(LMRN.NE.0) NINMAX=NINMAX-1
      IF(NINMAX.GE.2) GO TO 65
      WRITE(LUN,943)
  943 FORMAT('*0  ERROR IS DETECTED IN S/P "SHAPE".')
      STOP
C
   65 DO 70 NIN=1,NINMAX
      IF(NIN.EQ.NINMAX) NODINO=NB1+1
      INNB=NIN
      INNE=INSHMX-NIN
      JNNB=NIN
      JNNE=JNSHMX-NIN
      DO 72 I=INNB,INNE
      NB1=NB1+1
      IL(I,JNNB)=NB1
      XG(NB1)=FLOAT(I-INO)*DX
      YG(NB1)=FLOAT(JNNB-JNO)*DY
   72 CONTINUE
      DO 74 J=JNNB,JNNE
      NB1=NB1+1
      IL(INNE+1,J)=NB1
      XG(NB1)=FLOAT(INNE+1-INO)*DX
      YG(NB1)=FLOAT(J-JNO)*DY
   74 CONTINUE
      DO 76 I=INNB,INNE
      II=INNE+1+INNB-I
      NB1=NB1+1
      IL(II,JNNE+1)=NB1
      XG(NB1)=FLOAT(II-INO)*DX
      YG(NB1)=FLOAT(JNNE+1-JNO)*DY
   76 CONTINUE
      DO 78 J=JNNB,JNNE
      JJ=JNNB+JNNE+1-J
      NB1=NB1+1
      IL(INNB,JJ)=NB1
      XG(NB1)=FLOAT(INNB-INO)*DX
      YG(NB1)=FLOAT(JJ-JNO)*DY
   78 CONTINUE
   70 CONTINUE
      INNB=INNB+1
      JNNB=JNNB+1
      JJNB=JNNB+1
      JJNE=JNNE-1
C
      IF(LMREF.NE.0) GO TO 84
      DO 80 I=INNB,INNE
      DO 80 J=JNNB,JNNE
      IF(ICH(I,J).LE.-1) GO TO 80
      IF(ICH(I+1,J).EQ.-3.OR.ICH(I+1,J+1).EQ.-3.OR.
     1   ICH(I,J+1).EQ.-3.OR.ICH(I-1,J+1).EQ.-3.OR.
     2   ICH(I-1,J).EQ.-3.OR.ICH(I-1,J-1).EQ.-3.OR.
     3   ICH(I,J-1).EQ.-3.OR.ICH(I+1,J-1).EQ.-3) ICH(I,J)=-4
   80 CONTINUE
      GO TO 92
C
   84 DO 85 I=INNB,INNE
      DO 85 J=JNNB,JNNE
      IF(ICH(I,J).LE.-1) GO TO 85
      IF(ICH(I+1,J).LE.-1.OR.ICH(I+1,J+1).LE.-1.OR.
     1   ICH(I,J+1).LE.-1.OR.ICH(I-1,J+1).LE.-1.OR.
     2   ICH(I-1,J).LE.-1.OR.ICH(I-1,J-1).LE.-1.OR.
     3   ICH(I,J-1).LE.-1.OR.ICH(I+1,J-1).LE.-1) ICH(I,J)=-4
   85 CONTINUE
```

```
C
      DO 90 I=INNS,INNE
      DO 90 J=JNNS,JNNE
      IF(ICH(I,J).LE.-1) GO TO 90
      IF(ICH(I+1,J).LE.-1.OR.ICH(I+1,J+1).LE.-1.OR.
     1   ICH(I,J+1).LE.-1.OR.ICH(I-1,J+1).LE.-1.OR.
     2   ICH(I-1,J).LE.-1.OR.ICH(I-1,J-1).LE.-1.OR.
     3   ICH(I,J-1).LE.-1.OR.ICH(I+1,J-1).LE.-1) ICH(I,J)=1
   90 CONTINUE
C
   92 DO 95 I=INNS,INNE
      DO 95 J=JNNS,JNNE
      JJ=ICH(I,J)
      IF(JJ.LE.-1.OR.JJ.EQ.1) GO TO 95
      MS1=MS1+1
      IL(I,J)=MS1
      XG(MS1)=FLOAT(I-IHO)*DX
      YG(MS1)=FLOAT(J-JHO)*DY
   95 CONTINUE
      NODINS=MS1+1
C
      DO 100 I=INNS,INNE
      DO 110 J=JNNS,JNNE
      J1=ICH(I,J)
      IF(J1.EQ.-1) GO TO 120
      IF(J1.EQ.0) GO TO 110
      MS1=MS1+1
      IL(I,J)=MS1
      XG(MS1)=FLOAT(I-IHO)*DX
      YG(MS1)=FLOAT(J-JHO)*DY
      GO TO 130
  120 IL(I,J)=-1
  130 IF(I.EQ.INNS.OR.I.EQ.INNE) GO TO 110
      IF(J.EQ.JNNS.OR.J.EQ.JNNE) GO TO 110
      I1=ICK(I-1,J)
      I2=ICK(I,J)
      IF(I1.GE.5.AND.I1.LE.8) GO TO 140
      GO TO 110
  140 IF(I2.GE.5.AND.I2.LE.8) GO TO 150
      GO TO 110
  150 MS1=MS1+1
      XG(MS1)=FLOAT(I-IHO)*DX
      YG(MS1)=(FLOAT(J-JHO)+0.5)*DY
      MI=MI+1
      M=(MI-1)*3
      IBI(M+1)=I
      IBI(M+2)=J
      IBI(M+3)=MS1
  110 CONTINUE
C
      IF(I.EQ.INNE) GO TO 100
C
      DO 160 J=JJNS,JJNE
      J1=ICK(I,J-1)
      J2=ICK(I,J)
      IF(J1.GE.9.AND.J1.LE.12) GO TO 170
      GO TO 160
  170 IF(J2.GE.9.AND.J2.LE.12) GO TO 180
      GO TO 160
  180 MS1=MS1+1
      XG(MS1)=(FLOAT(I-IHO)+0.5)*DX
      YG(MS1)=FLOAT(J-JHO)*DY
      MJ=MJ+1
      M=(MJ-1)*3
      IBJ(M+1)=I
      IBJ(M+2)=J
      IBJ(M+3)=MS1
  160 CONTINUE
  100 CONTINUE
      MAXNOD=MS1
      KB(1)=NODOUT+1
      KE(1)=MS1
      KAM(1)=KE(1)-KB(1)+1
C
      WRITE(LUN,123) MAXNOD
  123 FORMAT('** MAXNOD=',I4)
C
      K=KELOUT
      DO 200 I=1,INGMIN
      DO 210 J=1,JMGMIN
      MMM=ICK(I,J)
      IF(MMM.EQ.-1) GO TO 210
      K=K+1
      IF(MMM.GT.0) GO TO 220
C
      IG(K,1)=IL(I,J)
      IG(K,2)=IL(I+1,J)
      IG(K,3)=IL(I+1,J+1)
      IG(K,4)=IL(I,J+1)
      GO TO 270
C
  220 IF(MMM.GE.5.AND.MMM.LE.8) GO TO 280
      IF(MMM.GE.9.AND.MMM.LE.12) GO TO 360
C
      GO TO (230,240,250,260),MMM
C
  230 IG(K,1)=IL(I,J)
      IG(K,2)=IL(I+1,J)
      IG(K,3)=IL(I+1,J+1)
      GO TO 270
  240 IG(K,1)=IL(I,J)
      IG(K,2)=IL(I+1,J)
      IG(K,3)=IL(I,J+1)
      GO TO 270
  250 IG(K,1)=IL(I,J)
      IG(K,2)=IL(I+1,J+1)
      IG(K,3)=IL(I,J+1)
      GO TO 270
  260 IG(K,1)=IL(I+1,J)
      IG(K,2)=IL(I+1,J+1)
      IG(K,3)=IL(I,J+1)
  270 ICK(I,J)=-1
      GO TO 210
C
  280 DO 290 M=1,MI
      MM=(M-1)*3
      IF(IBI(MM+1).NE.I+1) GO TO 290
      IF(IBI(MM+2).NE.J) GO TL 290
      IS=IBI(MM+3)
      GO TO 300
  290 CONTINUE
  300 MI=MMM-4
      GO TO(310,320,330,340),MI
C
  310 IG(K,1)=IL(I,J)
      IG(K,2)=IL(I+1,J)
      IG(K,3)=IS
      K=K+1
      IG(K,1)=IL(I+1,J)
      IG(K,2)=IL(I+2,J)
      IG(K,3)=IS
      K=K+1
      IG(K,1)=IS
      IG(K,2)=IL(I+2,J)
      IG(K,3)=IL(I+2,J+1)
      GO TO 350
  320 IG(K,1)=IL(I,J)
      IG(K,2)=IS
      IG(K,3)=IL(I,J+1)
      K=K+1
      IG(K,1)=IL(I,J)
      IG(K,2)=IL(I+1,J)
      IG(K,3)=IS
      K=K+1
      IG(K,1)=IL(I+1,J)
      IG(K,2)=IL(I+2,J)
      IG(K,3)=IS
      GO TO 350
  330 IG(K,1)=IL(I,J)
      IG(K,2)=IS
      IG(K,3)=IL(I,J+1)
      K=K+1
      IG(K,1)=IL(I,J+1)
      IG(K,2)=IS
      IG(K,3)=IL(I+1,J+1)
      K=K+1
      IG(K,1)=IL(I+1,J+1)
      IG(K,2)=IS
      IG(K,3)=IL(I+2,J+1)
      GO TO 350
  340 IG(K,1)=IL(I,J+1)
      IG(K,2)=IS
      IG(K,3)=IL(I+1,J+1)
      K=K+1
      IG(K,1)=IL(I+1,J+1)
      IG(K,2)=IS
      IG(K,3)=IL(I+2,J+1)
      K=K+1
      IG(K,1)=IL(I+2,J+1)
      IG(K,2)=IS
      IG(K,3)=IL(I+2,J)
  350 ICK(I,J)=-1
      ICK(I+1,J)=-1
      GO TO 210
C
  360 DO 370 M=1,MJ
      MM=(M-1)*3
      IF(IBJ(MM+1).NE.I) GO TO 370
      IF(IBJ(MM+2).NE.J+1) GO TO 370
      IS=IBJ(MM+3)
      GO TO 380
  370 CONTINUE
C
  380 MI=MMM-8
      GO TO (390,400,410,420),MI
C
  390 IG(K,1)=IL(I,J)
      IG(K,2)=IL(I+1,J)
      IG(K,3)=IS
      K=K+1
      IG(K,1)=IS
      IG(K,2)=IL(I+1,J)
      IG(K,3)=IL(I+1,J+1)
      K=K+1
      IG(K,1)=IS
      IG(K,2)=IL(I+1,J+1)
      IG(K,3)=IL(I+1,J+2)
      GO TO 430
  400 IG(K,1)=IL(I,J)
      IG(K,2)=IL(I+1,J)
      IG(K,3)=IS
      K=K+1
      IG(K,1)=IL(I,J)
      IG(K,2)=IS
      IG(K,3)=IL(I,J+1)
      K=K+1
      IG(K,1)=IL(I,J+1)
      IG(K,2)=IS
      IG(K,3)=IL(I,J+2)
      GO TO 430
  410 IG(K,1)=IL(I,J)
      IG(K,2)=IS
      IG(K,3)=IL(I,J+1)
      K=K+1
      IG(K,1)=IL(I,J+1)
      IG(K,2)=IS
      IG(K,3)=IL(I,J+2).
      K=K+1
      IG(K,1)=IL(I,J+2)
      IG(K,2)=IS
      IG(K,3)=IL(I+1,J+2)
      GO TO 430
  420 IG(K,1)=IL(I+1,J)
      IG(K,2)=IL(I+1,J+1)
      IG(K,3)=IS
      K=K+1
      IG(K,1)=IS
      IG(K,2)=IL(I+1,J+1)
      IG(K,3)=IL(I+1,J+2)
      K=K+1
      IG(K,1)=IS
      IG(K,2)=IL(I+1,J+2)
      IG(K,3)=IL(I,J+2)
  430 ICK(I,J)=-1
      ICK(I,J+1)=-1
C
  210 CONTINUE
  200 CONTINUE
      KMAX=K-KELOUT
      MAXELE=K
C
C
      MM=1
      IF(MM.EQ.1) GO TO 1302
      DO 1301 J=1,JMGMAX
      JJ=JMGMAX+1-J
      WRITE(LUN,1350) (ICH(I,JJ),I=1,INGMAX)
 1301 CONTINUE
      ICASE=1
      IF(INGMAX.GT.25) ICASE=2
      DO 440 II=1,ICASE
      IF(II.EQ.2) GO TO 450
      I1=1
      I2=MIN0(INGMAX,25)
      GO TO 460
  450 I1=25
      I2=INGMAX
      WRITE(LUN,979)
  979 FORMAT(////)
  460 DO 470 J=1,JMGMAX
      J1=JMGMAX+1-J
      WRITE(LUN,982) (IL(I,J1),I=I1,I2)
  982 FORMAT(25I5)
  470 CONTINUE
```

```
  440 CONTINUE
 1502 CONTINUE
      WRITE(LUN,973) NMAX,NODINO,NODINS
  973 FORMAT(' ** # OF CELLS IN AN INNERMOST NEST=',I5/
     1 ' ** NODINO=',I5/' ** NODINS=',I5)
      DO 500 I=1,IMSHMX
      DO 500 J=1,JMSHMX
      JJ=ICH(I,J)
      IF(JJ.EQ.1.OR.JJ.EQ.-4) ICH(I,J)=0
  500 CONTINUE
C
      CALL EXEC(1,LU,IBUF,64,ITRACS+2,55)
      IBUF(26)=NODINO
      IBUF(27)=NODINS
      IBUF(28)=NODINS-NODINO
      CALL EXEC(2,LU,IBUF,64,ITRACS+2,55)
      CALL EXEC(8,NAME)
C
      STOP
      END
      END$


FTN4
$EMA(AEMA1,0)
      PROGRAM ASSIN(5,150)
C
C  THIS S/P DOES RENUMBERING OF NODE# AT THE SURFACE OF OBJECT
C   SO THAT LAST 'NSURF' NODE# ARE ALWAYS CORRESPONDING TO THE
C   SURFACE NODE#, WHERE 'NSURF' IS A TOTAL # OF SURFACE NODE.
C
      COMMON/AEMA1/XG(2560),YG(2560),IG(2560,4),IL(53,53),
     1ICH(53,53),ICK(52,52)
      COMMON/CMESH/IMSHIN,JMSHIN,IMSHMX,JMSHMX,IHO,JHO,DX,DY
      COMMON/CNST2/NESMAX,MAXNOD,MAXELE,NODOUT,NODIN,KELOUT,
     1KELIN,NODINS
      COMMON/IODAT/LUN,LU,ITRACS
      COMMON/OBJ1/NOBJCT,NSEGO(5),ISPACE(64),JSPACE(64),XSSEG(64),
     1YSSEG(64),XESEG(64),YESEG(64),ISLP(64),IVEC(64),SLPSEG(64),
     2CGOX(5),CGOY(5)
      COMMON/OBJ2/XSP(128),YSP(128),XEP(128),YEP(128),NSP(128),
     1NEP(128),NNP(32),NPAT(5),MAXPAT
      COMMON/BLK1/XS(15,6),YS(15,6),XE(15,6),YE(15,6),TY(15,7),
     1YYM1(15,5),YYM2(15,5)
      COMMON/BLK4/IBUF1(25),IBUF2(25),NODE(25),IBUF4(25)
      DIMENSION NSURF(128),IBUF(64),NAME(3),IAUX(128)
      EQUIVALENCE (ICK,IAUX)
C     DATA NAME/2HSH,2HAD,2HE /
      DATA NAME/2HOL,2HNT,2HI /
C
      IFLG=0
      WRITE(LUN,999)
  999 FORMAT(//20('*'),5X,'"ASSIN" STARTS',5X,20('*'))
      MASA=0
      IF(MASA.EQ.0) GO TO 690
      ICASE=1
      IF(IMSHMX.GT.25) ICASE=2
      DO 600 II=1,ICASE
      IF(II.EQ.2) GO TO 610
      I1=1
      I2=MINO(IMSHMX,25)
      GO TO 620
  610 I1=25
      I2=IMSHMX
      WRITE(LUN,979)
  979 FORMAT(///)
  620 DO 630 J=1,JMSHMX
      J1=JMSHMX+1-J
      WRITE(LUN,982) (IL(I,J1),I=I1,I2)
  982 FORMAT(25I5)
  630 CONTINUE
  600 CONTINUE
C
      WRITE(LUN,670) ((IG(I,J),J=1,4),I=KELIN,MAXELE)
      WRITE(LUN,680) (I,XG(I),YG(I),I=NODIN,MAXNOD)
  670 FORMAT(24I5)
  680 FORMAT(8(I4,2F6.2))
  690 IPAT=0
      LA=NODINS
      LB=MAXNOD
      MAXSEG=NSEGO(NOBJCT)
      DO 1000 N=1,MAXSEG
      M=0
      IS=ISLP(N)
      IV=IVEC(N)
      HI1=XESEG(N)
      HI2=XSSEG(N)
      HJ1=YESEG(N)
      HJ2=YSSEG(N)
      IF(IS.EQ.0) GO TO 40
      IF(IS.EQ.3) GO TO 60
      G=FLOAT(IS)/2.0
      IF(IV.EQ.1) GO TO 20
      IF(IS.LT.0) GO TO 10
      DO 100 L=LA,LB
      IF(XG(L).GE.HI1.AND.XG(L).LE.HI2) GO TO 110
      GO TO 100
  110 IF(YG(L).GE.HJ1.AND.YG(L).LE.HJ2) GO TO 120
      GO TO 100
  120 IF(ABS(G*(XG(L)-HI1)+HJ1-YG(L)).LE.0.01*DY) GO TO 125
      GO TO 100
  125 M=M+1
      NODE(M)=L
  100 CONTINUE
      NZ=1
      GO TO 400
   10 DO 130 L=LA,LB
      IF(XG(L).GE.HI1.AND.XG(L).LE.HI2) GO TO 140
      GO TO 130
  140 IF(YG(L).GE.HJ2.AND.YG(L).LE.HJ1) GO TO 145
      GO TO 130
  145 IF(ABS(G*(XG(L)-HI1)+HJ1-YG(L)).LE.0.01*DY) GO TO 147
      GO TO 130
  147 M=M+1
      NODE(M)=L
  130 CONTINUE
      NZ=2
      GO TO 400
   20 IF(IS.GT.0) GO TO 30
      DO 150 L=LA,LB
      IF(XG(L).GE.HI2.AND.XG(L).LE.HI1) GO TO 160
      GO TO 150
  160 IF(YG(L).GE.HJ1.AND.YG(L).LE.HJ2) GO TO 170
      GO TO 150
  170 IF(ABS(G*(XG(L)-HI1)+HJ1-YG(L)).LE.0.01*DY) GO TO 175
      GO TO 150
  175 M=M+1
      NODE(M)=L
  150 CONTINUE
      NZ=3
      GO TO 400
   30 DO 180 L=LA,LB
      IF(XG(L).GE.HI2.AND.XG(L).LE.HI1) GO TO 185
      GO TO 180
  185 IF(YG(L).GE.HJ2.AND.YG(L).LE.HJ1) GO TO 190
      GO TO 180
  190 IF(ABS(G*(XG(L)-HI1)+HJ1-YG(L)).LE.0.01*DY) GO TO 195
      GO TO 180
  195 M=M+1
      NODE(M)=L
  180 CONTINUE
      NZ=4
      GO TO 400
   40 IF(IV.EQ.1) GO TO 50
      DO 200 L=LA,LB
      IF(YG(L).EQ.HJ1) GO TO 210
      GO TO 200
  210 IF(XG(L).GE.HI1.AND.XG(L).LE.HI2) GO TO 220
      GO TO 200
  220 M=M+1
      NODE(M)=L
  200 CONTINUE
      NZ=5
      GO TO 400
   50 DO 250 L=LA,LB
      IF(YG(L).EQ.HJ1) GO TO 260
      GO TO 250
  260 IF(XG(L).GE.HI2.AND.XG(L).LE.HI1) GO TO 270
      GO TO 250
  270 M=M+1
      NODE(M)=L
  250 CONTINUE
      NZ=6
      GO TO 400
   60 IF(IV.EQ.1) GO TO 70
      DO 300 L=LA,LB
      IF(XG(L).EQ.HI1) GO TO 310
      GO TO 300
  310 IF(YG(L).GE.HJ1.AND.YG(L).LE.HJ2) GO TO 320
      GO TO 300
  320 M=M+1
      NODE(M)=L
  300 CONTINUE
      NZ=7
      GO TO 400
   70 DO 350 L=LA,LB
      IF(XG(L).EQ.HI1) GO TO 360
      GO TO 350
  360 IF(YG(L).GE.HJ2.AND.YG(L).LE.HJ1) GO TO 370
      GO TO 350
  370 M=M+1
      NODE(M)=L
  350 CONTINUE
      NZ=8
  400 MMAX=M
      DO 410 M=1,MMAX
      IBUF2(M)=M
      IBUF4(M)=NODE(M)
      IF(NZ.GE.7) GO TO 420
      IBUF1(M)=XG(IBUF4(M))*1000.0
      GO TO 410
  420 IBUF1(M)=YG(IBUF4(M))*1000.0
  410 CONTINUE
C
      CALL SORT(MMAX)
C
      L=(MMAX+1)*(1-IV)/2
      DO 430 M=1,MMAX
      LL=L+IV*M
      NODE(M)=IBUF4(IBUF2(LL))
  430 CONTINUE
      MMAX=MMAX-1
      DO 440 M=1,MMAX
      IPAT=IPAT+1
      NSP(IPAT)=NODE(M)
      NEP(IPAT)=NODE(M+1)
  440 CONTINUE
      IF(IFLG.EQ.0) GO TO 1000
      NNP(N)=IPAT
      DO 450 M=1,NOBJCT
      IF(N.NE.NSEGO(M)) GO TO 450
      IF(M.EQ.1) GO TO 460
      NPAT(M)=IPAT-NPAT(M-1)
      GO TO 450
  460 NPAT(M)=IPAT
  450 CONTINUE
 1000 CONTINUE
      IF(IFLG.EQ.1) GO TO 480
      IAUX(1)=IPAT
      DO 470 I=1,IPAT
      IAUX(I+1)=NSP(I)
  470 CONTINUE
      IFLG=1
      CALL RSURF
      GO TO 690
  480 MAXPAT=IPAT
      NSURF=MAXPAT
      DO 490 M=1,MAXPAT
      L=NSP(M)
      LL=NEP(M)
      NSURF(M)=L
      XSP(M)=XG(L)
      YSP(M)=YG(L)
      XEP(M)=XG(LL)
      YEP(M)=YG(LL)
  490 CONTINUE
      NODFS=MAXNOD-NSURF
C
      CALL EXEC(1,LU,IBUF,64,ITRACS+2,55)
      IBUF(9)=NODFS
      IBUF(10)=MAXELE
      IBUF(11)=MAXNOD
      IBUF(12)=MAXPAT
      IBUF(13)=NSURF
      CALL EXEC(2,LU,IBUF,64,ITRACS+2,55)
      CALL EXEC(2,LU,XSP,256,ITRACS+3,0)
      CALL EXEC(2,LU,YSP,256,ITRACS+3,8)
      CALL EXEC(2,LU,XEP,256,ITRACS+3,16)
      CALL EXEC(2,LU,YEP,256,ITRACS+3,24)
      CALL EXEC(2,LU,NSP,128,ITRACS+3,32)
      CALL EXEC(2,LU,NEP,128,ITRACS+3,36)
      CALL EXEC(2,LU,NSURF,128,ITRACS+3,40)
      DO 510 I=1,32
      IBUF(I)=NNP(I)
  510 CONTINUE
      DO 520 I=1,5
      IBUF(I+32)=NPAT(I)
  520 CONTINUE
      CALL EXEC(2,LU,IBUF,64,ITRACS+3,44)
      CALL EXEC(8,NAME)
C
```

```
      STOP
      END
$EMA(AEMA1,0)
      SUBROUTINE RSURF
      COMMON/AEMA1/XG(2560),YG(2560),IG(2560,4),IL(53,53),
     1ICH(53,53),ICK(52,52)
      COMMON/CMESH/IMSHIN,JMSHIN,IMSHMX,JMSHMX,IHO,JHO,DX,DY
      COMMON/CNST2/NESMAX,MAXNOD,MAXELE,NODOUT,NODIN,KELOUT,
     1KELIN,NODINS
      DIMENSION MSURF(128),BUFX(200),BUFY(200),NAME(3)
      EQUIVALENCE (ICK(1,1),MSURF),(ICK(2,1),NSURF)
C
      N1=MSURF-1
      M=1
    1 M1=M+1
    2 IF(NSURF(M).LE.NSURF(M1)) GO TO 3
      I1=NSURF(M)
      NSURF(M)=NSURF(M1)
      NSURF(M1)=I1
    3 M1=M1+1
      IF(M1.LE.MSURF) GO TO 2
      M=M+1
      IF(M.LE.N1) GO TO 1
C
      DO 100 M=1,MSURF
      MM=NSURF(M)
      BUFX(M)=XG(MM)
      BUFY(M)=YG(MM)
      DO 120 I=1,IMSHMX
      DO 120 J=1,JMSHMX
      LL=IL(I,J)
      IF(LL.EQ.MM) GO TO 140
  120 CONTINUE
      GO TO 160
  140 IL(I,J)=-LL
  160 DO 180 I=KELIN,MAXELE
      DO 180 J=1,4
      LL=IG(I,J)
      IF(LL.EQ.MM) IG(I,J)=-LL
  180 CONTINUE
  100 CONTINUE
C
      DO 200 M=1,MSURF
      M1=NSURF(M)+1
      IF(M.NE.MSURF) M2=NSURF(M+1)-1
      IF(M.EQ.MSURF) M2=MAXNOD
      MSHIFT=M
      IF(M1.GT.M2) GO TO 200
      DO 220 MM=M1,M2
      NEW=MM-MSHIFT
      XG(NEW)=XG(MM)
      YG(NEW)=YG(MM)
      DO 240 I=1,IMSHMX
      DO 240 J=1,JMSHMX
      LL=IL(I,J)
      IF(LL.EQ.MM) GO TO 260
  240 CONTINUE
      GO TO 280
  260 IL(I,J)=NEW
  280 DO 300 I=KELIN,MAXELE
      DO 300 J=1,4
      LL=IG(I,J)
      IF(LL.EQ.MM) IG(I,J)=NEW
  300 CONTINUE
  220 CONTINUE
  200 CONTINUE
C
      MO=MAXNOD-MSURF
      DO 320 M=1,MSURF
      NEW=MO+M
      XG(NEW)=BUFX(M)
      YG(NEW)=BUFY(M)
      MM=NSURF(M)
      DO 340 I=1,IMSHMX
      DO 340 J=1,JMSHMX
      LL=IL(I,J)
      IF(LL.GE.-1) GO TO 340
      IF(IABS(LL).EQ.MM) GO TO 360
  340 CONTINUE
      GO TO 380
  360 IL(I,J)=NEW
  380 DO 400 I=KELIN,MAXELE
      DO 400 J=1,4
      LL=IG(I,J)
      IF(LL.GE.0) GO TO 400
      IF(IABS(LL).EQ.MM) IG(I,J)=NEW
  400 CONTINUE
  320 CONTINUE
      RETURN
      END
      END$



FTN4
$EMA(AEMA1,0)
      PROGRAM MESTR(5,150)
C
C THIS S/P GENERATES A MESH OF RECTANGULAR FINITE ELEMENTS
C  OUTSIDE INNEMOST ONE AND DEFINE THE NUMBERS OF THE FOUR NODES
C  OF EACH ELEMENT.
C
      COMMON/AEMA1/XG(2560),YG(2560),IG(2560,4),IL(53,53),
     1ICH(53,53),ICK(52,52)
      COMMON/CMESH/IMSHIN,JMSHIN,IMSHMX,JMSHMX,IHO,JHO,DX,DY
      COMMON/CNST1/NNEST(10),NNI(10),NNJ(10),NOUTER(10),KS(10),
     1KE(10),KAM(10),KAMA(10),MBW(10),MBWA(10)
      COMMON/CNST2/NESMAX,MAXNOD,MAXELE,NODOUT,NODIN,KELOUT,
     1KELIN,NODINS
      COMMON/CONN/NCONN(10,4,4)
      COMMON/IODAT/LUN,LU,ITRACS
      COMMON/PSOLV/ISYM,IPOISS,IDPOT,RDEBY,CPOIS
      COMMON/BLK4/IBUF1(25),IBUF2(25),IBUF3(25),IBUF4(25)
      DIMENSION BNSTX(10,2),BNSTY(10,2),BUF(64),IBUF(64),NAME(3)
      DIMENSION JBUF(256),NCONA(4,48),NINNER(200)
      EQUIVALENCE(JBUF,BUF,IBUF),(ICK,NINNER)
      DATA NAME/2HME,2HMA,2H1 /
C
      WRITE(LUN,999)
  999 FORMAT(//20'.',5X, 'MESTR' STARTS',5X,20'.')
      MS=0
      INES=NESMAX-1
      XB1=XG(IL(IMSHMX,JMSHMX))
      YB1=YG(IL(IMSHMX,JMSHMX))
      BNSTX(1,1)= XB1
      BNSTX(1,2)=XB1
      BNSTY(1,1)=-YB1
      BNSTY(1,2)=YB1
      NN=2
      IF(ISYM.EQ.1) NN=1
      DO 40 I=1,INES
      J1=NESMAX+I-1
      XB=XB1
      YB=YB1
      DO 60 J=2,J1
      XB=XB+DX*FLOAT(2**(J-1)*NNEST(J))
      YB=YB+DY*FLOAT(2**(J-1)*NNEST(J))
   60 CONTINUE
      BNSTX(J1,1)=-XB
      BNSTX(J1,2)=XB
      BNSTY(J1,1)=-YB
      BNSTY(J1,2)=YB
      NI=NNI(J1)
      NJ=NNJ(J1)
C
      IF(J1.LT.NESMAX) GO TO 75
      IF(ISYM.EQ.1) GO TO 72
      NOUTER(1)=1
      NOUTER(2)=NI
      NOUTER(3)=NI+NJ-1
      NOUTER(4)=NI*2+NJ-2
      NOUTER(5)=(NI+NJ)*2-4
      DO 65 J=1,5
      NOUTER(J+5)=NOUTER(J)+J-1
   65 CONTINUE
      GO TO 75
   72 NOUTER(2)=1
      NOUTER(3)=NJ
      NOUTER(4)=NI+NJ-1
      NOUTER(5)=NOUTER(4)
C
   75 J2=NNEST(J1)
      HX=DX*FLOAT(2**(J1-1))
      HY=DY*FLOAT(2**(J1-1))
      KS(J1)=MS+1
      DO 50 J=1,J2
      X1=XB-HX*FLOAT(J-1)
      Y1=YB-HY*FLOAT(J-1)
      NN=2
      IF(ISYM.EQ.1) NN=1
      N1=NI-NN*(J-1)
      N2=NJ-NN*(J-1)
      IF(ISYM.EQ.1) N1=N1+1
      IF(ISYM.EQ.1) GO TO 85
      DO 80 N=1,N1
      MS=MS+1
      XG(MS)=-X1+HX*FLOAT(N-1)
      YG(MS)=-Y1
   80 CONTINUE
   85 DO 90 N=1,N2
      MS=MS+1
      XG(MS)=X1
      YG(MS)=-Y1*FLOAT(ISYM/3)+HY*FLOAT(N-1)
   90 CONTINUE
      DO 100 N=1,N1
      MS=MS+1
      XG(MS)=X1-HX*FLOAT(N-1)
      YG(MS)=Y1
  100 CONTINUE
      IF(ISYM.EQ.1) GO TO 50
      DO 110 N=1,N2
      MS=MS+1
      XG(MS)=-X1
      YG(MS)=Y1-HY*FLOAT(N-1)
  110 CONTINUE
   50 CONTINUE
      KE(J1)=MS
      KAM(J1)=KE(J1)-KS(J1)+1
   40 CONTINUE
C
C     WRITE(LUN,125) (N,XG(N),YG(N),N=1,MAXNOD)
C 125 FORMAT(7(I5,2F6.2))
      WRITE(LUN,127) (I,KS(I),KE(I),KAM(I),I=1,NESMAX)
  127 FORMAT('**',18X,'  KS   KE  KAM'/(4X,'NEST # =',I2,5X,3I5))
      WRITE(LUN,128) (NOUTER(I),I=1,5)
  128 FORMAT('**  NOUTER',5X,5I5)
C
      M=0
      DO 600 I=1,10
      DO 600 J=1,2
      M=M+1
      BUF(M)=BNSTX(I,J)
      BUF(M+32)=BNSTY(I,J)
  600 CONTINUE
      CALL EXEC(2,LU,BUF,128,ITRACS+4,64)
      CALL EXEC(1,LU,IBUF,64,ITRACS+2,55)
      DO 610 I=1,10
      IBUF(I+30)=NOUTER(I)
  610 CONTINUE
      CALL EXEC(2,LU,IBUF,64,ITRACS+2,55)
      DO 620 I=1,NESMAX
      IBUF(I)=KS(I)
      IBUF(I+10)=KE(I)
      IBUF(I+20)=KAM(I)
  620 CONTINUE
      CALL EXEC(2,LU,IBUF,64,ITRACS+4,66)
C
      M=0
      M1=0
      IF(ISYM.EQ.1) GO TO 142
      DO 140 I=1,IMSHMX,2
      M=M+1
      NINNER(M)=IL(I,1)
      IF(I.EQ.IMSHMX) GO TO 140
      M1=M1+1
      NCONA(1,M1)=IL(I+1,1)
  140 CONTINUE
      GO TO 145
  142 M=M+1
      NINNER(M)=IL(IMSHMX,1)
  145 M1=0
      DO 150 J=3,JMSHMX,2
      M=M+1
      M1=M1+1
      NINNER(M)=IL(IMSHMX,J)
      NCONA(2,M1)=IL(IMSHMX,J-1)
  150 CONTINUE
      M1=0
      DO 160 I=3,IMSHMX,2
      M=M+1
      M1=M1+1
      I1=IMSHMX+1-I
      NINNER(M)=IL(I1,JMSHMX)
      NCONA(3,M1)=IL(I1+1,JMSHMX)
  160 CONTINUE
      IF(ISYM.EQ.1) GO TO 172
      J1=JMSHMX-2
      M1=0
      DO 170 J=3,J1,2
      M=M+1
      M1=M1+1
      J2=JMSHMX+1-J
```

```
      NINNER(M)=IL(1,J2)
      NCONA(4,M1)=IL(1,J2+1)
  150 CONTINUE
      M1=M1+1
      NCONA(4,M1)=IL(1,2)
  152 MINN=MAX0((IMSHMX-1)/2,(JMSHMX-1)/2)
      WRITE(LUN,985) (NINNER(I),I=1,M)
  985 FORMAT(' ** NINNER '(20I5))
      WRITE(LUN,986) ((NCONA(I,J),J=1,MINN),I=1,4)
  986 FORMAT(' ** NCONA'/(20I5))

      K=0
      MS=1
      IF(ISYM.EQ.1) MS=0
      DO 200 I=1,INES
      I1=NESMAX+1-I
      NI=NNI(I1)
      NJ=NNJ(I1)
      IF(ISYM.EQ.1) GO TO 240
      K=K+1
      IG(K,1)=MS
      IG(K,2)=MS+1
      IG(K,3)=MS+(NI+NJ)*2
      IG(K,4)=IG(K,3)-1
  240 J1=MNEST(I1)
      DO 250 J=1,J1
      NN=2
      IF(ISYM.EQ.1) NN=1
      M1=NI-NN*J
      M2=NJ-NN*J
      IF(I.EQ.INES.AND.J.EQ.J1) GO TO 340
      IF(J.EQ.J1) GO TO 270
      JUMP=1
      GO TO 280
  270 JUMP=2
  280 IF(J.EQ.1) GO TO 290
C
      K=K+1
      IF(ISYM.EQ.1) GO TO 292
      IG(K,1)=IG(K-1,1)+1
      IG(K,2)=IG(K,1)+1
      IG(K,3)=IG(K-1,3)+1
      IG(K,4)=IG(K,3)-1
      GO TO 290
  292 IG(K,1)=IG(K-1,1)+1
      IG(K,2)=IG(K-1,4)+1
      IG(K,3)=IG(K,2)+1
      IG(K,4)=IG(K,1)+JUMP
      IF(J.NE.J1) GO TO 307
      NCONN(I1,2,1)=K
      NCONN(I1,2,3)=IG(K,1)+1
      GO TO 307
  290 IF(ISYM.EQ.1) GO TO 306
      IF(J.NE.J1) GO TO 295
      NCONN(I1,1,1)=K+1
      NCONN(I1,1,3)=IG(K,3)+1
C
  295 DO 300 M=1,M1
      K=K+1
      IG(K,1)=IG(K-1,1)+1
      IG(K,2)=IG(K-1,2)+1
      IG(K,3)=IG(K-1,3)+JUMP
      IG(K,4)=IG(K,3)-JUMP
  300 CONTINUE
      IF(J.NE.J1) GO TO 305
      NCONN(I1,1,2)=K
      NCONN(I1,1,4)=IG(K,4)+1
  305 K=K+1
      IG(K,1)=IG(K-1,1)+1
      IG(K,2)=IG(K,1)+1
      IG(K,3)=IG(K,2)+1
      IG(K,4)=IG(K-1,3)
      IF(J.NE.J1) GO TO 308
      NCONN(I1,2,1)=K+1
      NCONN(I1,2,3)=IG(K,4)+1
      GO TO 308
  306 K=K+1
      IG(K,1)=MS+2+NI+NJ
      IG(K,2)=MS+1
      IG(K,3)=MS+2
      IG(K,4)=IG(K,1)+JUMP
      IF(J.NE.J1) GO TO 307
      NCONN(I1,2,1)=K
      NCONN(I1,2,3)=IG(K,1)+1
  307 M2=M2-1
C
  308 DO 310 M=1,M2
      K=K+1
      IG(K,1)=IG(K-1,4)
      IG(K,2)=IG(K-1,2)+1
      IG(K,3)=IG(K-1,3)+1
      IG(K,4)=IG(K-1,4)+JUMP
  310 CONTINUE
      IF(J.NE.J1) GO TO 315
      NCONN(I1,2,2)=K
      NCONN(I1,2,4)=IG(K,1)+1
  315 K=K+1
      IG(K,1)=IG(K-1,4)
      IG(K,2)=IG(K-1,2)+1
      IG(K,3)=IG(K,2)+1
      IG(K,4)=IG(K,3)+1
      IF(J.NE.J1) GO TO 318
      NCONN(I1,3,1)=K+1
      NCONN(I1,3,3)=IG(K,1)+1
C
  318 DO 320 M=1,M1
      K=K+1
      IG(K,1)=IG(K-1,1)+JUMP
      IG(K,2)=IG(K,1)-JUMP
      IG(K,3)=IG(K-1,3)+1
      IG(K,4)=IG(K,3)+1
  320 CONTINUE
      IF(J.NE.J1) GO TO 325
      NCONN(I1,3,2)=K
      NCONN(I1,3,4)=IG(K,2)+1
  325 IF(ISYM.EQ.1) GO TO 250
      K=K+1
      IG(K,1)=IG(K-1,4)+2
      IG(K,2)=IG(K-1,2)+JUMP
      IG(K,3)=IG(K-1,3)+1
      IG(K,4)=IG(K,3)+1
      IF(J.NE.J1) GO TO 328
      NCONN(I1,4,1)=K+1
      NCONN(I1,4,3)=IG(K,2)+1
C
  328 M2=M2-1
      DO 330 M=1,M2
      K=K+1
      IG(K,1)=IG(K-1,1)+1
      IG(K,2)=IG(K-1,2)+JUMP
      IG(K,3)=IG(K-1,2)
      IG(K,4)=IG(K-1,4)+1
  330 CONTINUE
      K=K+1
      IG(K,1)=IG(K-1,1)+1
      IG(K,2)=IG(K,1)+1
      IG(K,3)=IG(K-1,2)
      IG(K,4)=IG(K-1,4)+1
      IF(J.NE.J1) GO TO 250
      NCONN(I1,4,2)=K
      NCONN(I1,4,4)=IG(K,3)+1
  250 CONTINUE
      MS=IG(K,2)
      IF(ISYM.EQ.1) MS=IG(K,4)
  200 CONTINUE
C
  340 M=0
      DO 630 I=1,4
      DO 630 J=1,48
      M=M+1
      JBUF(M)=NCONA(I,J)
  630 CONTINUE
      CALL EXEC(2,LU,JBUF,256,ITRACS+4,54)
C
      IF(ISYM.EQ.1) GO TO 352
      MS=IG(K,2)
      K=K+1
      IG(K,1)=MS
      IG(K,2)=MS+1
      IG(K,3)=NINNER(1)
      IG(K,4)=IG(K,3)-1
      NCONN(I1,1,1)=K+1
      DO 350 M=1,M1
      K=K+1
      IG(K,1)=IG(K-1,2)
      IG(K,2)=IG(K,1)+1
      IG(K,3)=NINNER(M+1)
      IG(K,4)=IG(K-1,3)
  350 CONTINUE
      NCONN(I1,1,2)=K
      K=K+1
      IG(K,1)=IG(K-1,2)
      IG(K,2)=IG(K,1)+1
      IG(K,3)=IG(K,2)+1
      IG(K,4)=IG(K-1,3)
      NCONN(I1,2,1)=K+1
C
  352 M4=M1+1
      IF(ISYM.EQ.4) GO TO 355
      K=K+1
      IG(K,1)=NINNER(1)
      IG(K,2)=IG(K-1,4)+1
      IG(K,3)=IG(K,2)+1
      IG(K,4)=NINNER(2)
      NCONN(I1,2,1)=K
      M4=2
      M2=M2-1
  355 DO 360 M=1,M2
      K=K+1
      M3=M4+M
      IG(K,1)=IG(K-1,4)
      IG(K,2)=IG(K-1,3)
      IG(K,3)=IG(K,2)+1
      IG(K,4)=NINNER(M3)
  360 CONTINUE
      NCONN(I1,2,2)=K
      K=K+1
      IG(K,1)=IG(K-1,4)
      IG(K,2)=IG(K-1,3)
      IG(K,3)=IG(K,2)+1
      IG(K,4)=IG(K,3)+1
      NCONN(I1,3,1)=K+1
C
      M4=M1+M2+1
      IF(ISYM.EQ.1) M4=M2+2
      DO 370 M=1,M1
      K=K+1
      M3=M4+M
      IG(K,1)=NINNER(M3)
      IG(K,2)=IG(K-1,1)
      IG(K,3)=IG(K-1,4)
      IG(K,4)=IG(K,3)+1
  370 CONTINUE
      NCONN(I1,3,2)=K
      IF(ISYM.EQ.1) GO TO 382
      K=K+1
      IG(K,1)=IG(K-1,4)+2
      IG(K,2)=IG(K-1,1)
      IG(K,3)=IG(K-1,4)
      IG(K,4)=IG(K,3)+1
      NCONN(I1,4,1)=K+1
C
      M4=M1*2+M2+1
      M2=M2-1
      DO 380 M=1,M2
      K=K+1
      M3=M4+M
      IG(K,1)=IG(K-1,1)+1
      IG(K,2)=NINNER(M3)
      IG(K,3)=IG(K-1,2)
      IG(K,4)=IG(K-1,1)
  380 CONTINUE
      K=K+1
      IG(K,1)=IG(K-1,1)+1
      IG(K,2)=NINNER(1)
      IG(K,3)=IG(K-1,2)
      IG(K,4)=IG(K-1,1)
      NCONN(I1,4,2)=K
  382 KELOUT=K
C     WRITE(LUN,385) ((IG(K,M),M=1,4),K=1,KELOUT)
C 385 FORMAT(20I5)
      WRITE(LUN,987) ((I,J,(NCONN(I,J,K),K=1,4),J=1,4),I=2,NESMAX)
  987 FORMAT('0*  NCONN'/2X,'  I     J    K=1    K=2    K=3    K=4'/
     A(2I5,4I6))
C
      K1=KELOUT+1
C     WRITE(LUN,395) ((IG(K,M),M=1,4),K=K1,MAXELE)
C 395 FORMAT(20I5)
C     RETURN
C
      M=0
      DO 640 I=1,10
      DO 640 J=1,4
      DO 640 K=1,4
      M=M+1
      JBUF(M)=NCONN(I,J,K)
  640 CONTINUE
      CALL EXEC(2,LU,JBUF,256,ITRACS+4,58)
      CALL EXEC(8,NAME)
C
      STOP
      END
      END$
```

```
FTN4
$EMA(AEMA1,0)
      PROGRAM WEMA1(5,150)
C
C     THIS S/P WRITES EMA PARAMETERS ON DISCS.
C
      COMMON/AEMA1/XG(2560),YG(2560),IG(2560,4),IL(53,53),
     1ICH(53,53),ICN(52,52)
      COMMON/CMESH/IMSHIN,JMSHIN,IMSHMX,JMSHMX,IHO,JHO,DX,DY
      COMMON/IODAT/LUN,LU,ITRACS
      DIMENSION IBUF1(256),IBUF2(256),BUF1(128),BUF2(128)
      EQUIVALENCE (IBUF1,BUF1),(IBUF2,BUF2)
C
      WRITE(LUN,999)
  999 FORMAT(//20'.',5X,'"WEMA1" STARTS',5X,20'.')
      NSTOP=3072/128
      ISTART=0
      ISECT=0
      DO 100 N=1,NSTOP
      DO 110 I=1,128
      II=ISTART+I
      BUF1(I)=XG(II)
      BUF2(I)=YG(II)
  110 CONTINUE
      CALL EXEC(2,LU,BUF1,256,ITRACS+8,ISECT)
      CALL EXEC(2,LU,BUF2,256,ITRACS+9,ISECT)
      ISECT=ISECT+4
      ISTART=ISTART+128
  100 CONTINUE
C
      NSTOP=2560/256
      ISECT=0
      DO 200 J=1,2
      M=J+2
      ISTART=0
      DO 210 N=1,NSTOP
      DO 220 I=1,256
      II=ISTART+I
      IBUF1(I)=IG(II,J)
      IBUF2(I)=IG(II,M)
  220 CONTINUE
      CALL EXEC(2,LU,IBUF1,256,ITRACS+10,ISECT)
      CALL EXEC(2,LU,IBUF2,256,ITRACS+11,ISECT)
      ISECT=ISECT+4
      ISTART=ISTART+256
  210 CONTINUE
  200 CONTINUE
C
      DO 250 I=1,IMSHMX
      IBUF1(I)=IL(I,1)
      IBUF1(I+64)=IL(I,JMSHMX)
  250 CONTINUE
      DO 260 J=1,JMSHMX
      IBUF1(J+128)=IL(1,J)
      IBUF1(J+192)=IL(IMSHMX,J)
  260 CONTINUE
      CALL EXEC(2,LU,IBUF1,256,ITRACS+2,81)
      IPRO=IMSHMX*JMSHMX
      II=0
      M=0
      ISECT=0
      DO 300 I=1,IMSHMX
      DO 310 J=1,JMSHMX
      II=II+1
      M=M+1
      IBUF1(M)=IL(I,J)
      IBUF2(M)=ICH(I,J)
      IF(II.EQ.IPRO) GO TO 320
      IF(M.LT.256) GO TO 310
  320 CALL EXEC(2,LU,IBUF1,256,ITRACS+12,ISECT)
      CALL EXEC(2,LU,IBUF2,256,ITRACS+13,ISECT)
      ISECT=ISECT+4
      M=0
  310 CONTINUE
  300 CONTINUE
C
      WRITE(LUN,781)
  781 FORMAT(///)
      STOP
      END
      END$
```

```
FTN4
$EMA(AEMA2,0)
      PROGRAM GLMAT(5,150)
C
C THIS IS THE MAIN ROUTINE FOR THE GLOBAL MATRIX  SECTION
C  OF  SYCIE .
C
      COMMON/AEMA2/AM(4096),JA(4096),IA(512),XG(2560),YG(2560),
     1IG(2560,4),YBC(512),JBC(512),IBC(512),NCONU(128),
     2NCONV(128),NCONN(128),NCONX(256)
      COMMON/CNST1/NOUTER(10)
      COMMON/CNST2/NESMAX,MAXNOD,MAXELE,NODFS,NODINS
      COMMON/CONN/NCONU,NCONA(4,48),NCONN(10,4,4)
      COMMON/ELEM/SXA(10),SYA(10),SMA(10),SXB(10),SYB(10),
     1SMB(10),SXC1(21),SYC1(21),SMC1(21),SXC2(10),SYC2(10),
     2SMC2(10),SXC3(15),SYC3(15),SMC3(15),SXD(10),SYD(10),
     3SMD(10),SXE1(15),SYE1(15),SME1(15),SXE2(28),SYE2(28),
     4SME2(28),SXF(36),SYF(36),SMF(36)
      COMMON/IODAT/LUN,LU,ITRACS
      COMMON/PSOLV/ISYM,IPOISS,IBPOT,RDEBY,CPOIS
      COMMON/BLK1/ELMAT(9,9)
      COMMON/BLK4/IBUF1(25),IBUF2(25),IGA(25),IGG(25)
      COMMON/BLK6/A1,A2,A3,DENOM,AA1,AA2
      COMMON/BLK7/NOMB,NVAR,JPMAX,NBC,K,K1,K2,K3,KK,J,NES,M
      COMMON/BLK8/NND,NZ,NZ1,NZ2,NZ3,NCTYP
      DIMENSION NAME(3)
      DATA NAME/2HID,2HAT,2H2 /
C
      WRITE(LUN,999)
  999 FORMAT(//20'.',5X,'"GLMAT" STARTS',5X,20'.')
      CALL EXEC(8,NAME)
      STOP
      END
      BLOCK DATA
      COMMON/CNST1/NOUTER(10)
      COMMON/CNST2/NESMAX,MAXNOD,MAXELE,NODFS,NODINS
      COMMON/CONN/NCONU,NCONA(4,48),NCONN(10,4,4)
      COMMON/ELEM/SXA(10),SYA(10),SMA(10),SXB(10),SYB(10),
     1SMB(10),SXC1(21),SYC1(21),SMC1(21),SXC2(10),SYC2(10),
     2SMC2(10),SXC3(15),SYC3(15),SMC3(15),SXD(10),SYD(10),
     3SMD(10),SXE1(15),SYE1(15),SME1(15),SXE2(28),SYE2(28),
     4SME2(28),SXF(36),SYF(36),SMF(36)
      COMMON/IODAT/LUN,LU,ITRACS
      COMMON/PSOLV/ISYM,IPOISS,IBPOT,RDEBY,CPOIS
      COMMON/BLK1/ELMAT(9,9)
      COMMON/BLK4/IBUF1(25),IBUF2(25),IGA(25),IGG(25)
      COMMON/BLK6/A1,A2,A3,DENOM,AA1,AA2
      COMMON/BLK7/NOMB,NVAR,JPMAX,NBC,K,K1,K2,K3,KK,J,NES,M
      COMMON/BLK8/NND,NZ,NZ1,NZ2,NZ3,NCTYP
      DATA LUN/6/,LU/51/, ITRACS/0/
      DATA NVAR/0/,JPMAX/0/,NBC/0/
      END
      SUBROUTINE SORT(N)
      COMMON/BLK4/IAA(25),IBB(25),IBUF3(25),IBUF4(25)
C
      N1=N-1
      M=1
    1 M1=M+1
    2 IF(IAA(M).LE.IAA(M1)) GO TO 3
      I1=IAA(M)
      I2=IBB(M)
      IAA(M)=IAA(M1)
      IBB(M)=IBB(M1)
      IAA(M1)=I1
      IBB(M1)=I2
    3 M1=M1+1
      IF(M1.LE.N) GO TO 2
      M=M+1
      IF(M.LE.N1) GO TO 1
      RETURN
      END
      END$
```

```
FTN4
$EMA(AEMA2,0)
      PROGRAM GLMT2(5,150)
C
C THIS S/P FORMS GLOBAL MATRIX FOR POTENTIAL SOLVER AND
C  APPLIES BOUNDARY CONDITIONS.
C
      COMMON/AEMA2/AM(4096),JA(4096),IA(512),XG(2560),YG(2560),
     1IG(2560,4),YBC(512),JBC(512),IBC(512),NCONU(128),
     2NCONV(128),NCONN(128),NCONX(256)
      COMMON/CNST1/NOUTER(10)
      COMMON/CNST2/NESMAX,MAXNOD,MAXELE,NODFS,NODINS
      COMMON/CONN/NCONU,NCONA(4,48),NCONN(10,4,4)
      COMMON/IODAT/LUN,LU,ITRACS
      COMMON/PSOLV/ISYM,IPOISS,IBPOT,RDEBY,CPOIS
      COMMON/BLK1/ELMAT(9,9)
      COMMON/BLK4/IBUF1(25),IBUF2(25),IGA(25),IGG(25)
      COMMON/BLK6/A1,A2,A3,DENOM,AA1,AA2
      COMMON/BLK7/NOMB,NVAR,JPMAX,NBC,K,K1,K2,K3,KK,J,NES,M
      COMMON/BLK8/NND,NZ,NZ1,NZ2,NZ3,NCTYP
      DIMENSION AYO(256),BOFF(2,32),BDIA(2,32),NAME(3)
      EQUIVALENCE (XG,AYO)
      DATA NAME/2HWE,2HMA,2H2 /
C
      WRITE(LUN,999)
  999 FORMAT(//20'.',5X,'"GLMT2" STARTS',5X,20'.')
      MANC=1
      IF(MANC.EQ.0) GO TO 333
      NES1=MAX0(NESMAX-1,1)
      DO 160 NES=NES1,NESMAX
      KK=NESMAX+1-NES
      K1=1
      IF(KK.NE.NESMAX) K1=NCONN(KK+1,4,2)+1
      K2=MAXELE
      IF(KK.NE.1) K2=NCONN(KK,1,1)-1
      WRITE(LUN,971) NES,KK,K1,K2
  971 FORMAT(' ** NES=',I2,5X,'KK=',I2,5X,'K1=',I4,5X,'K2=',I4)
      DO 170 K=K1,K2
      IG1=IG(K,1)
      IG2=IG(K,2)
      IG3=IG(K,3)
      IF(IG(K,4).EQ.0) GO TO 220
C
      NND=4
      HX=XG(IG2)-XG(IG1)
      HY=YG(IG3)-YG(IG2)
      A1=HY/HX
      A2=HX/HY
      A3=HX*HY*CPOIS
C
      IF(KK.NE.1) GO TO 210
      IF(NCONU.EQ.0) GO TO 210
      DO 190 I=1,NCONU
      II=I
```

```
      IF(NCONN(I).EQ.K) GO TO 200
  190 CONTINUE
      GO TO 210
  200 NCTYP=NCONV(II)
      CALL FELMT
      GO TO 170
C
  210 NCTYP=0
      CALL FELMT
      GO TO 170
C
  220 NND=3
      IGA(1)=IG1
      IGA(2)=IG2
      IGA(3)=IG3
      HA=(XG(IG2)*YG(IG3)+XG(IG1)*YG(IG2)+
     1XG(IG3)*YG(IG1))-(XG(IG2)*YG(IG1)+XG(IG3)*
     2YG(IG2)+XG(IG1)*YG(IG3))
      B1=YG(IG2)-YG(IG3)
      B2=YG(IG3)-YG(IG1)
      B3=YG(IG1)-YG(IG2)
      A1=XG(IG3)-XG(IG2)
      A2=XG(IG1)-XG(IG3)
      A3=XG(IG2)-XG(IG1)
      C1=HA**2/12.0*CPOIS
      C2=HA*2.0
      DENOM=C2
      ELMAT(1,1)=(B1**2+A1**2-2.0*C1)
      ELMAT(1,2)=(B2*B1+A2*A1-C1)
      ELMAT(1,3)=(B3*B1+A3*A1-C1)
      ELMAT(2,2)=(B2**2+A2**2-2.0*C1)
      ELMAT(2,3)=B3*B2+A3*A2-C1
      ELMAT(3,3)=B3**2+A3**2-2.0*C1
C
  230 CALL FILE
  170 CONTINUE
C
  235 CONTINUE
      WRITE(LUN,973) KK
  973 FORMAT('     KK=',I2)
      IF(KK.EQ.1) GO TO 160
      DO 240 J=1,4
      K1=NCONN(KK,J,1)
      K2=NCONN(KK,J,2)
      NZ=J+2-J/3*4
      IF(KK.NE.2) K3=NCONN(KK,J,3)
      M=0
      DO 250 K=K1,K2
      M=M+1
      IF(KK.NE.2) IGG(5)=K3+(M-1)*2
      IF(KK.EQ.2) IGG(5)=NCONA(J,M)
      NCTYP=-1
      CALL FELMT
  250 CONTINUE
      IF(J.EQ.4) GO TO 240
      K=K2+1
      NCTYP=0
      CALL FELMT
  240 CONTINUE
  160 CONTINUE
C
      IA(NVAR+1)=JPMAX+1
C
      DO 270 N=1,NVAR
      I1=IA(N)
      I2=IA(N+1)-1
      NO=NODINS+N-1
      DO 280 I=I1,I2
      II=I
      IF(JA(I).EQ.NO) GO TO 290
  280 CONTINUE
      WRITE(LUN,980)
  980 FORMAT('ERROR IS DETECTED')
      STOP
  290 IF(II.EQ.I1) GO TO 270
      NN=JA(II)
      A=AM(II)
      I3=I1+1
      DO 300 I=I3,II
      IO=I3+II-I
      JA(IO)=JA(IO-1)
      AM(IO)=AM(IO-1)
  300 CONTINUE
      JA(I1)=NN
      AM(I1)=A
  270 CONTINUE
C
C******APPLY THE BOUNDARY CONDITIONS
C  IBPOT IS THE FLAG INDICATING THE TYPE OF BOUNDARY CONDITIONS
C   IMPOSED ON THE OUTERMOST EDGES.
C     IBPOT=1: PHI=0.0
C     IBPOT=2: PHI=k/R
C     IBPOT=3: PHI=k*EXP(-R/RDEBY)
C     IBPOT=4: PHI=k*EXP(-R/RDEBY)/R
C
C     WRITE(LUN,568) K,(AM(IA(I)),I=1,64)
  333 IF(IBPOT.EQ.1) GO TO 510
      NOMB=0
      DEBA=RDEBY*2.0
      NND=4
      DENOM=1.0
C     DO 330 M=1,4
      DO 330 M=1,2
      K1=NOUTER(M)
      K2=NOUTER(M+1)
C     IF(M.NE.4) K2=K2-1
C     WRITE(LUN,567) M,K1,K2
C 567 FORMAT('M=',I3,5X,'K1=',I3,5X,'K2=',I3)
      L=0
      DO 340 K=K1,K2
      L=L+1
      NZ=M
      DO 350 I=1,4
      IGA(I)=IO(K,I)
      DO 350 J=1,4
      ELMAT(I,J)=0.0
  350 CONTINUE
      IG1=IGA(1)
      IG2=IGA(2)
      IG3=IGA(3)
      IG4=IGA(4)
      HX=XG(IG2)-XG(IG1)
      HY=YG(IG3)-YG(IG2)
C
      NFLG=0
      IF(K.EQ.K1.OR.K.EQ.K2) NFLG=1
C     IF(K.EQ.K1) NFLG=1
  360 GO TO (370,380,390,400),NZ
  370 B1=XG(IG1)
      B2=HX
      B3=ABS(YG(IG1))
      GO TO 410
  380 B1=YG(IG2)
      B2=HY
      B3=ABS(XG(IG2))
      GO TO 410
  390 B1=XG(IG4)
      B2=HX
      B3=ABS(YG(IG4))
      GO TO 410
  400 B1=YG(IG1)
      B2=HY
      B3=ABS(XG(IG1))
  410 E1=B2**2
      IF(IBPOT.EQ.3) GO TO 420
      D1=ATAN(B2*B3/(B1**2+B3**2+B1*B2))
      D2=ALOG(1.0+B2*(2.0*B1+B2)/(B1**2+B3**2))
      P1=(((B1+B2)**2-B3**2)*D1+(B2-(B1+B2)*D2)*B3)
      P2=(((B1+B2/2.0)*D2-B2)*B3+(B3**2-B1**2-B1*B2)*D1)
      P3=((B1**2-B3**2)*D1+(B2-B1*D2)*B3)
      IF(IBPOT.EQ.2) GO TO 440
      IF(IBPOT.EQ.4) GO TO 430
  420 P1=0.0
      P2=0.0
      P3=0.0
  430 B4=SQRT(1.0+((B1+B2)/B3)**2)
      B5=SQRT(1.0+(B1/B3)**2)
      B6=B2/(B4+B5)*((B2+2.0*B1)/B3)
      B7=B2*B3*B4
      D3=ALOG(1.0+(B2+B6)/(B1+B5*B3))
      P1=P1+(B7-(3.0*B1+4.0*B2)*B6+(2.0*(B1-B2)**2-B3**2)*D3)
     1/DEBA
      P2=P2+((3.0*B1+2.0*B2)*B6-B7+(2.0*B1*(B1-B2)-B3**2)*D3)
     1/DEBA
      P3=P3+(B7-3.0*B1*B6+(2.0*B1**2-B3**2)*D3)/DEBA
  440 P1=P1/E1
      P2=P2/E1
      P3=P3/E1
C
      M1=NZ
      M2=NZ+1-NZ/4*4
      IF(NZ.LE.2) GO TO 450
      M1=M2
      M2=NZ
  450 ELMAT(M1,M1)=ELMAT(M1,M1)+P1
      ELMAT(M1,M2)=ELMAT(M1,M2)+P2
      ELMAT(M2,M2)=ELMAT(M2,M2)+P3
      IF(NZ.EQ.3) ELMAT(3,4)=ELMAT(4,3)
C     WRITE(LUN,888) L,P1,P2,P3
C 888 FORMAT('L=',I3,5X,'P1=',1PE12.5,5X,'P2=',1PE12.5,
C    15X,'P3=',1PE12.5)
      IF(NFLG.EQ.0) GO TO 500
      NFLG=0
      IF(K.EQ.K1) NZ=(NZ-1)+(5-NZ)/4*4
      IF(K.EQ.K2) NZ=(NZ+1)-NZ/4*4
      GO TO 360
C
  500 CONTINUE
      IF(K.NE.K1) BDIA(M,L)=BDIA(M,L)+ELMAT(M,M)
      IF(K.EQ.K1) BDIA(M,L)=ELMAT(M,M)
      IF(K.NE.K2) BDIA(M,L+1)=BDIA(M,L+1)+ELMAT(M+1,M+1)
      IF(K.EQ.K2) BDIA(M,L+1)=ELMAT(M+1,M+1)
      BOFF(M,L)=ELMAT(M,M+1)
C     WRITE(LUN,507) K,(IGA(I),I=1,4)
C 507 FORMAT(5X,'K=',I3,5X,'IGA=',4I6)
C     WRITE(LUN,509) ((ELMAT(I,J),J=1,4),I=1,4)
C 509 FORMAT(1P4E13.5)
C     CALL FILE
C     WRITE(LUN,568) K,(AM(IA(I)),I=1,64)
C 568 FORMAT('*K=',I3/(1P10E13.5))
  340 CONTINUE
  330 CONTINUE
      WRITE(LUN,732)
  732 FORMAT('** BDIA')
      WRITE(LUN,733) ((BDIA(M,L),L=1,20),M=1,2)
  733 FORMAT(1P10E13.5)
      WRITE(LUN,777)
  777 FORMAT('** BOFF')
      WRITE(LUN,734) ((BOFF(M,L),L=1,20),M=1,2)
  734 FORMAT(1P10E13.5)
      DO 1000 I=1,NVAR
      I1=IA(I)
      I2=IA(I+1)-1
      WRITE(LUN,1001) I,(JA(J),J=I1,I2)
 1001 FORMAT(I3,3X,12I9)
      WRITE(LUN,1002) (AM(J),J=I1,I2)
 1002 FORMAT(6X,12F9.4)
 1000 CONTINUE
      CALL EXEC(2,LU,BDIA,128,ITRACS+4,81)
      CALL EXEC(2,LU,BOFF,128,ITRACS+4,83)
C     STOP
      GO TO 800
C
  510 NOMB=NOUTER(5)
      NA=NOMB+1
      DO 600 N=NA,NODFS
      IP1=IA(N)
      IP2=IA(N+1)-1
      DO 610 I=IP1,IP2
      JP=JA(I)
      IF(JP.GT.NOMB) GO TO 610
      JA(I)=-1
      LAST=N
  610 CONTINUE
  600 CONTINUE
      NC=1
      IA(1)=1
      NUM=0
      DO 650 N=NA,LAST
      IP1=IA(N)
      IP2=IA(N+1)-1
      DO 660 I=IP1,IP2
      JP=JA(I)
      IF(JP.EQ.-1) GO TO 660
      NUM=NUM+1
      JA(NUM)=JP
      AM(NUM)=AM(I)
  660 CONTINUE
      NC=NC+1
      IA(NC)=NUM+1
  650 CONTINUE
      NB=IP2+1
      NB=LAST+1
      NSUB=IA(NB)-IA(NC)
      NB=NB+1
      DO 700 N=NB,NODFS
      NC=NC+1
      IA(NC)=IA(N)-NSUB
  700 CONTINUE
      NVAR=NC
      DO 710 N=NB,JPMAX
      I=N-NSUB
      AM(I)=AM(N)
      JA(I)=JA(N)
```

```
   10 CONTINUE
      JEMAX=JEMAX+NSUB
      IA(NVAR+1)=JEMAX+1
      DO 250 N=1,NBC
      IBC(N)=IBC(N)+NUMB
  250 CONTINUE
C
  800 DO 850 N=1,NVAR
      A(N)=0.0
  850 CONTINUE
      CALL EXEC(6,NAME)
      STOP
      END
$EMA(AEMA2,0)
      SUBROUTINE FELMT
      COMMON/AEMA2/AM(4096),JA(4096),IA(512),XG(2560),YG(2560),
     1IG(2560,4),YBC(512),JBC(512),IBC(512),NCONU(128),
     2NCONV(128),NCONW(128),NCONX(256)
      COMMON/CONN/MCONU,NCONA(4,48),NCONN(10,4,4)
      COMMON/ELEM/SXA(10),SYA(10),SMA(10),SXB(10),SYB(10),
     1SMB(10),SXC1(21),SYC1(21),SMC1(21),SXC2(10),SYC2(10),
     2SMC2(10),SXC3(15),SYC3(15),SMC3(15),SXD(10),SYD(10),
     3SMD(10),SXE1(15),SYE1(15),SME1(15),SXE2(28),SYE2(28),
     4SME2(28),SXF(36),SYF(36),SMF(36)
      COMMON/IODAT/LUN,LU,ITRACS
      COMMON/BLK1/ELMAT(9,9)
      COMMON/BLK4/IBUF1(25),IBUF2(25),IGA(25),IGG(25)
      COMMON/BLK6/A1,A2,A3,DENOM,AA1,AA2
      COMMON/BLK7/NOMB,NVAR,JPMAX,NBC,K,ID6,ID7,ID8,ID9,ID10,
     1ID11,ID12
      COMMON/BLK8/NND,NZ,NZ1,NZ2,NZ3,NCTYP
C
      DO 100 I=1,4
      IGG(I)=IG(K,I)
  100 CONTINUE
      IF(NCTYP.EQ.0) GO TO 470
      IF(NCTYP.GE.1) GO TO 110
      MTYP=1
      GO TO 130
  110 IF(NCTYP.LE.4) MTYP=1
      IF(NCTYP.GE.5.AND.NCTYP.LE.10) MTYP=2
      IF(NCTYP.GE.11.AND.NCTYP.LE.14) MTYP=3
      IF(NCTYP.EQ.15) MTYP=4
      MM=NCONU(N)
      DO 120 I=1,MTYP
      I1=MM-1+I
      IGG(I+4)=NCONX(I1)
  120 CONTINUE
      IF(NCTYP.EQ.9.OR.NCTYP.EQ.10) GO TO 300
      IF(MTYP.EQ.4) GO TO 410
      NZ=NCTYP-(MTYP+2)*(MTYP-1)
  130 NZ1=NZ+1-NZ/4*4
      NZ2=NZ+2-NZ/3*4
      NZ3=NZ-1+(5-NZ)/4*4
      IF(NZ.EQ.2.OR.NZ.EQ.4) GO TO 140
      AA1=A1
      AA2=A2
      GO TO 150
  140 AA1=A2
      AA2=A1
  150 GO TO (160,190,350,300,410,470),MTYP
  160 NND=4
      DENOM=144.0
      DO 170 M=1,2
      N=0
      DO 180 I=1,4
      DO 180 J=I,4
      N=N+1
      ELMAT(I,J)=AA1*SXB(N)*24.0+AA2*SYB(N)*6.0-A3*SMB(N)
  180 CONTINUE
      IF(M.EQ.1) NZ4=NZ
      IF(M.EQ.2) NZ4=NZ1
      IGA(1)=IGG(NZ4)
      IGA(2)=IGG(5)
      IGA(M+2)=IGG(NZ2)
      IGA(5-M)=IGG(NZ3)
      CALL FILE
  170 CONTINUE
      RETURN
C
  190 DENOM=576.0
      NZ4=5
      IF(NZ.EQ.2.OR.NZ.EQ.3) NZ4=6
      DO 200 M=1,4
      N=0
      GO TO (210,230,250,260),M
C
  210 DO 220 I=1,6
      DO 220 J=I,6
      N=N+1
      ELMAT(I,J)=(AA1*SXC1(N)+AA2*SYC1(N))*12.0-A3*SMC1(N)
  220 CONTINUE
      NND=6
      IGA(1)=IGG(NZ)
      IGA(2)=IGG(NZ1)
      IGA(3)=IGG(NZ2)
      IGA(4)=IGG(NZ3)
      IGA(5)=IGG(5)
      IGA(6)=IGG(6)
      GO TO 290
  230 DO 240 I=1,4
      DO 240 J=I,4
      N=N+1
      ELMAT(I,J)=(AA1*SXC2(N)+AA2*SYC2(N))*12.0-A3*SMC2(N)
  240 CONTINUE
      NND=4
      IGA(1)=IGG(NZ)
      IGA(2)=IGG(5)
      IGA(3)=IGG(NZ2)
      IGA(4)=IGG(6)
      GO TO 290
C
  250 AA1=A1
      AA2=A2
      GO TO 270
  260 AA1=A2
      AA2=A1
  270 DO 280 I=1,5
      DO 280 J=I,5
      N=N+1
      ELMAT(I,J)=(AA1*SXC3(N)+AA2*SYC3(N))*12.0-A3*SMC3(N)
  280 CONTINUE
      NND=5
      IGA(1)=IGG(NZ)
      IGA(3)=IGG(NZ2)
      IF(M.EQ.1) NZ5=5-NZ2
      IF(M.EQ.2) NZ5=5-NZ
      IGA(4)=IGG(NZ5)
      IGA(3*M-7)=IGG(11-NZ4)
      IGA(14-3*M)=IGG(NZ4)
  290 CALL FILE
  200 CONTINUE
      RETURN
C
  300 DENOM=72.0
      NND=4
      NZ=NCTYP-8
      DO 310 M=1,2
      N=0
      IF(NZ.EQ.2) GO TO 320
      AA1=A1
      AA2=A2
      IGA(1)=IGG(4*M-3)
      IGA(2)=IGG(8-3*M)
      IGA(3)=IGG(9-3*M)
      IGA(4)=IGG(2*M+2)
      GO TO 330
  320 AA1=A2
      AA2=A1
      IGA(1)=IGG(5*M-4)
      IGA(2)=IGG(8-2*M)
      IGA(3)=IGG(7-2*M)
      IGA(4)=IGG(3*M-1)
  330 DO 340 I=1,4
      DO 340 J=I,4
      N=N+1
      ELMAT(I,J)=(AA1*SXD(N)*4.0+AA2*SYD(N))*6.0-A3*SMD(N)
  340 CONTINUE
      CALL FILE
  310 CONTINUE
      RETURN
C
  350 DENOM=576.0
      DO 360 M=1,4
      N=0
      IF(M.GE.3) GO TO 380
      DO 370 I=1,5
      DO 370 J=I,5
      N=N+1
      ELMAT(I,J)=(AA1*SXE1(N)+AA2*SYE1(N))*12.0-A3*SME1(N)
  370 CONTINUE
      NND=5
      IGA(M)=IGG(NZ)
      IGA(3-M)=IGG(NZ1)
      IGA(3)=IGG(5)
      IGA(M+3)=IGG(6)
      IGA(6-M)=IGG(7)
      GO TO 400
C
  380 DO 390 I=1,7
      DO 390 J=I,7
      N=N+1
      ELMAT(I,J)=(AA1*SXE2(N)+AA2*SYE2(N))*12.0-A3*SME2(N)
  390 CONTINUE
      NND=7
      IGA(M-2)=IGG(NZ)
      IGA(5-M)=IGG(NZ1)
      IGA(M)=IGG(NZ2)
      IGA(7-M)=IGG(NZ3)
      IGA(5)=IGG(5)
      IGA(M+3)=IGG(6)
      IGA(10-M)=IGG(7)
  400 CALL FILE
  360 CONTINUE
      RETURN
C
  410 DENOM=576.0
      NND=8
      DO 420 M=1,4
      N=0
      IF(M/2*2.EQ.M) GO TO 430
      AA1=A1
      AA2=A2
      GO TO 440
  430 AA1=A2
      AA2=A1
  440 DO 450 I=1,8
      DO 450 J=I,8
      N=N+1
      ELMAT(I,J)=(AA1*SXF(N)+AA2*SYF(N))*12.0-A3*SMF(N)
  450 CONTINUE
      IGA(1)=M
      IGA(5)=M+4
      DO 460 I=2,4
      II=IGA(I-1)
      IGA(I)=II+1-II/4*4
      IGA(I+4)=IGA(I)+4
  460 CONTINUE
      CALL FILE
  420 CONTINUE
      RETURN
C
  470 DENOM=36.0
      NND=4
      N=0
      DO 480 I=1,4
      DO 480 J=I,4
      N=N+1
      ELMAT(I,J)=(A1*SXA(N)+A2*SYA(N))*6.0-A3*SMA(N)
  480 CONTINUE
      DO 490 I=1,4
      IGA(I)=IGG(I)
  490 CONTINUE
      CALL FILE
C
      RETURN
      END
$EMA(AEMA2,0)
      SUBROUTINE FILE
C
      COMMON/AEMA2/AM(4096),JA(4096),IA(512),XG(2560),YG(2560),
     1IG(2560,4),YBC(512),JBC(512),IBC(512),NCONU(128),
     2NCONV(128),NCONW(128),NCONX(256)
      COMMON/CNST2/NESMAX,MAXNOD,MAXELE,NODFS,NODINS
      COMMON/IODAT/LUN,LU,ITRACS
      COMMON/BLK1/ELMAT(9,9)
      COMMON/BLK4/IBUF1(25),IBUF2(25),IGA(25),IGG(25)
      COMMON/BLK6/A1,A2,A3,DENOM,AA1,AA2
      COMMON/BLK7/NOMB,NVAR,JPMAX,NBC,ID5,ID6,ID7,ID8,ID9,ID10,
     1ID11,ID12
      COMMON/BLK8/NND,NZ,NZ1,NZ2,NZ3,NCTYP
      DIMENSION AUX(10),AAUX(10),IAUX(10),JAUX(10)
C
      DO 50 N=1,NND
      DO 50 M=N,NND
      ELMAT(N,M)=ELMAT(N,M)/DENOM
      IF(N.EQ.M) GO TO 50
      ELMAT(M,N)=ELMAT(N,M)
   50 CONTINUE
C
      DO 100 N=1,NND
      NO=IGA(N)
      IF(NO.LT.NODINS.OR.NO.GT.NODFS) GO TO 100
      NNO=NO+1-NODINS
```

```
      NN1=NNO+1
      NUM=0
      DO 110 M=1,NND
      MO=IGA(M)
C     IF(MO.LT.NO) GO TO 110
      NUM=NUM+1
      AUX(NUM)=ELMAT(N,M)
      IAUX(NUM)=MO
  110 CONTINUE
C
      M=0
      DO 112 I=1,NUM
      I1=IAUX(I)
      VAL=AUX(I)
      IF(I1.LE.NODES) GO TO 116
      IF(NBC.EQ.0) GO TO 114
      DO 113 K=1,NBC
      K1=K
      IF(IBC(K).EQ.NO.AND.JBC(K).EQ.I1) GO TO 115
  113 CONTINUE
  114 NBC=NBC+1
      IBC(NBC)=NO
      JBC(NBC)=I1
      YBC(NBC)=VAL
      GO TO 112
  115 YBC(K1)=YBC(K1)+VAL
      GO TO 112
  116 M=M+1
      JAUX(M)=I1
      AAUX(M)=VAL
  112 CONTINUE
      IF(M.EQ.0) GO TO 100
      DO 119 I=1,M
      IAUX(I)=JAUX(I)
      AUX(I)=AAUX(I)
  119 CONTINUE
      NUM=M
      JP1=IA(NNO)
      NSHIFT=NUM
      IF(NVAR.GT.NNO) GO TO 120
      JP2=JPMAX
      JP3=JP2+1
      IF(NVAR.LT.NNO) GO TO 300
      GO TO 150
  120 DO 130 K=NN1,NVAR
      K1=IA(K)
      IF(K1.NE.0) GO TO 140
  130 CONTINUE
  140 JP2=K1-1
      JP3=K1
      IF(JP1.EQ.0) GO TO 195
  150 DO 180 K=1,NUM
      K1=IAUX(K)
      DO 190 L=JP1,JP2
      JP=JA(L)
      IF(JP.EQ.K1) NSHIFT=NSHIFT-1
      IF(JP.GE.K1) GO TO 180
  190 CONTINUE
  180 CONTINUE
      IF(NSHIFT.EQ.0) GO TO 220
  195 DO 200 K=JP3,JPMAX
      K1=JPMAX+JP3-K
      K2=K1+NSHIFT
      AM(K2)=AM(K1)
      JA(K2)=JA(K1)
  200 CONTINUE
      DO 210 K=NN1,NVAR
      K1=IA(K)
      IF(K1.NE.0) IA(K)=K1+NSHIFT
  210 CONTINUE
      MAX=JPMAX+NSHIFT
      IF(JP1.EQ.0) GO TO 300
C
  220 DO 240 K=1,NUM
      K1=IAUX(K)
      DO 250 L=JP1,JP2
      LL=L
      JP=JA(L)
      IF(JP.EQ.K1) GO TO 290
      IF(JP.GT.K1) GO TO 260
  250 CONTINUE
      LL=LL+1
      GO TO 280
  260 DO 270 L=LL,JP2
      L1=JP2+LL-L
      AM(L1+1)=AM(L1)
      JA(L1+1)=JA(L1)
  270 CONTINUE
  280 AM(LL)=AUX(K)
      JA(LL)=K1
      JP2=JP2+1
      GO TO 240
  290 AM(LL)=AM(LL)+AUX(K)
  240 CONTINUE
      GO TO 340
C
  300 DO 310 K=1,NUM
      IBUF1(K)=IAUX(K)
      IBUF2(K)=K
  310 CONTINUE
      IF(NUM.GE.2) CALL SORT(NUM)
C
      DO 320 K=1,NUM
      K2=IBUF2(K)
      K1=JP3+K-1
      AM(K1)=AUX(K2)
      JA(K1)=IAUX(K2)
  320 CONTINUE
      IA(NNO)=JP3
  340 IF(NNO.GT.NVAR) NVAR=NNO
      JPMAX=JPMAX+NSHIFT
  100 CONTINUE
      IF(NVAR.GE.0) RETURN
      WRITE(LUN,887) ID5,(IDA(I),I=1,NND)
  887 FORMAT('CELL#=',I3,5X,8I5)
      WRITE(LUN,888) (IA(I),I=1,NVAR)
  888 FORMAT('#IA'/(25I5))
      WRITE(LUN,889) (JA(I),I=1,JPMAX)
  889 FORMAT('#JA'/(25I5))
      WRITE(LUN,901) (AM(I),I=1,JPMAX)
  901 FORMAT('#AM'/(1P10E13.5))
C
      RETURN
      END
      END$


FTN4
$FMA(AEMA2,0)
        FROM FORM AEMA2(5,150)
```

```
C
C THIS S/P WRITES FMA VARIABLES ON DISCS.
C
      COMMON/AEMA2/AM(4096),JA(4096),IA(512),XG(2560),YG(2560),
     1IG(2560,4),YBC(512),JBC(512),IBC(512),NCONU(128),
     2NCONV(128),NCONW(128),NCONX(256)
      COMMON/IODAT/LUN,LU,ITRACS
      COMMON/PSOLV/ISYM,IPOISS,IMPOT,KDEBY,LPOIS
      COMMON/BLK7/NOMB,NVAR,JPMAX,NBC,ID5,ID6,ID7,ID8,ID9,ID10,
     1ID11,ID12
      DIMENSION BUFR(128),IBUF1(128),IBUF2(128),JBUF(64),NAME(3)
      DIMENSION AYO(256)
      EQUIVALENCE(IBUF2,JBUF),(XG,AYO)
      DATA NAME/2HDC,2HOM,2HP /
C
      WRITE(LUN,999)
  999 FORMAT(//20'.',5X,'"WEMA2" STARTS',5X,20'.')
      CALL EXEC(1,LU,JBUF,64,ITRACS+2,55)
      JBUF(22)=NVAR
      JBUF(23)=JPMAX
      JBUF(24)=NBC
      JBUF(25)=NOMB
      CALL EXEC(2,LU,JBUF,64,ITRACS+2,55)
      WRITE(LUN,971) NVAR,JPMAX,NBC,NOMB
  971 FORMAT('##  NVAR=',I4,5X,'JPMAX= ',I5,5X,'NBC=',I4,5X,
     1'NOMB=',I4)
C
      ISECT=0
      ISTART=0
      ITRACK=ITRAC  30
      NSTOP=JPMAX/128+1
      DO 150 N=1,NSTOP
      DO 160 I=1,128
      IBUF1(I)=JA(ISTART+I)
  160 CONTINUE
      CALL EXEC(2,LU,IBUF1,128,ITRACK,ISECT)
      ISECT=ISECT+2
      ISTART=ISTART+128
      IF(MOD(N,48).NE.0) GO TO 150
      ITRACK=ITRACK+1
      ISECT=0
  150 CONTINUE
C
      ISECT=0
      ISTART=0
      ITRACK=ITRACS+14
      DO 170 N=1,NSTOP
      DO 180 I=1,128
      BUFR(I)=AM(ISTART+I)
  180 CONTINUE
      CALL EXEC(2,LU,BUFR,256,ITRACK,ISECT)
      ISECT=ISECT+4
      ISTART=ISTART+128
      IF(MOD(N,24).NE.0) GO TO 170
      ITRACK=ITRACK+1
      ISECT=0
  170 CONTINUE
C
      NSTOP=(NVAR+1)/128+1
      ISTART=0
      ISECT=0
      DO 200 N=1,NSTOP
      DO 210 I=1,128
      IBUF1(I)=IA(ISTART+I)
  210 CONTINUE
      CALL EXEC(2,LU,IBUF1,128,ITRACS+34,ISECT)
      ISECT=ISECT+2
      ISTART=ISTART+128
  200 CONTINUE
C
      NSTOP=NBC/128+1
      ISECT=0
      ISTART=0
      ITRACK=ITRACS+35
      DO 250 N=1,NSTOP
      DO 260 I=1,128
      IBUF1(I)=IBC(ISTART+I)
      IBUF2(I)=JBC(ISTART+I)
      BUFR(I)=YBC(ISTART+I)
  260 CONTINUE
      CALL EXEC(2,LU,IBUF1,128,ITRACK,ISECT)
      CALL EXEC(2,LU,IBUF2,128,ITRACK,ISECT+24)
      CALL EXEC(2,LU,BUFR,256,ITRACK,ISECT*2+48)
      ISECT=ISECT+2
      ISTART=ISTART+128
  250 CONTINUE
C     WRITE(LUN,972) (IBC(I),I=1,NBC)
C 972 FORMAT('IBC'/(25I5))
C     WRITE(LUN,974) (JBC(I),I=1,NBC)
C 974 FORMAT('JBC'/(25I5))
C
      NSTOP=NVAR/128+1
      ISTART=0
      ISECT=0
      DO 350 N=1,NSTOP
      DO 360 I=1,128
      BUFR(I)=AYO(ISTART+I)
  360 CONTINUE
      CALL EXEC(2,LU,BUFR,256,ITRACS+36,ISECT)
      ISECT=ISECT+4
      ISTART=ISTART+128
  350 CONTINUE
      IF(IPOISS.GT.0) STOP 0001
      IF(IPOISS.LT.0) CALL EXEC(8,NAME)
      STOP 0002
      END
      END$
```

```
FTN4
AEMA,AEMA...
      PROGRAM XYCIC(3,150)
C
C THIS IS THE MAIN ROUTINE FOR THE CURRENT COLLECTION SECTION
C   OF 'XYCIC'.
C
      DOUBLE PRECISION CCHB
      COMMON AEMA3 IG(2560,4),IWORK(256),POT(2560),EX(2560),
     1EY(2560),AM(5840),RWORK(256),CION(128),CELE(128),CPRD(128),
     2CFROM(128),CSEC(128),CSECM(128),CBSC(128),CBSCM(128),
     3CPHO(128),CPHOM(128),FSECT(128),CSECT(128),PEMIT(128),
     4XSP(128),YSP(128),XEP(128),YEP(128),XSSEG(64),YSSEG(64),
     5XESEG(64),YESEG(64),SLPSEG(64),IROOT(128),IOA(128),
     6NODEMT(128),NCONU(128),NCONV(128),NCONW(128),NCONX(128),
     7NEMTS(128),NEMTE(128),NSECT(128),NCOND(128),NPPAN(64,2),
     8NSUMP(128),ISLP(64),IVEC(64),MPAT(64),NCP(128),NCNA(128),
     9NCNB(384),NCNC(128),NSP(128),NEP(128),NSURF(128)
      COMMON/CACAG/TN,TJ1,TJ2,TX,TMIN,TMAX,ITN,IT1,IT2,ITX,II,
     1JJJ,JT,JJ1
      COMMON/CACPT/IK1,IDM1,IDM2,IDM3,IDM4,DM1
      COMMON/CACP1/T1,T2,T3,T4,T5,T6,TS,LA,LB,NA,NB,NNA,NNB,
     1IU,JU,MU
      COMMON/CGUES/IK2,NK2,IFL,JGS,JGE,NF,JSF,JEF,VAL,PHIF,PHINEW,
     1CURONL,IS1
      COMMON/CHEB/CCHB(31),IORDER
      COMMON/CHFLX/I,J,JJ,J1,J2,MAXD,COEF,RATE,DEN,ID,JD,ICUT,JCUT
      COMMON/CLFLX/IK3,JK3,JM,FF0,FF1,FF2,FF3,FF4
      COMMON/CLITG/F1,F2,F3,F4,F5,TL1,TL2,TL3,TL4,V1,V2,V3,V4,V5,
     1INTYP,ICASE,ITYPE
      COMMON/CMESH/DX,DY
      COMMON/CNST1/NNEST(10),BNSTX(10,2),BNSTY(10,2)
      COMMON/CNST2/NESMAX,MAXNOD,MAXELE,NODOUT,NODIN,KELOUT,KELIN
      COMMON/CONN/NCONU,NCONA(4,48),NCONN(10,4,4)
      COMMON/CORB/NEMTP,AGEMT,BETA,VMIN,POTS,BETAP,GAMA,NPTCL,
     1LPTCL,NCHA,IORB,JORB,JONL,PINF(10),PINFI(15),TRAP(2,7)
      COMMON/CRATO/RATIO(2,8),RTPR,RTSC,RTBC,RTPH,FLXREF
      COMMON/CROOT/ACURCY,DPHI
      COMMON/CSITG/AMIN(16),AMAX(16),VL(16),VA,VB,TM1,TM2,IITYPE,
     1ILEV,ISOL,IND,NLEV
      COMMON/CSUN/SANGL,NSUNB(32),XSUNB(32),YSUNB(32),ASUNS(64),
     1NSUN
      COMMON/CURT/CSUM,CARIV,CURE,TCUR,TCURM,SUM,CURR(3)
      COMMON/CUTAG/AACUT(5,4),NEMT,NCFLG,NNCUT,CUTMIN,CUTMAX
      COMMON/CVEL/VLEV(16),NLEV,NLEVI,VLEVI(16),VLEVE(16),
     1NLEVE,RANGEE(10),COEFTE(10),POWERE(10),IDISTF(2),
     2NPARTE,NDOM(16)
      COMMON/ECONN/NCFLG(4),NFIX(8),VFIX(8),NFLOT(64),
     1NBIAS(2,8),VBIAS(8),NRES(2,8),RESTA(8),ICOND(64)
      COMMON/EMITR/NCSEC(32),NPANC(64),NPANL,NEMIT,NEMS,NEME,
     1NINTP,NSECT,NSURF
      COMMON/GENRL/LUN3,ASPOT,ICNTRL,NTOUR,ITER,
     1ACTIME,TLINIT,IOBJF,ITIME(5)
      COMMON/IODAT/LUN1,LUN2,LU,ITRACS,NTRACK(32),NITER
      COMMON/MPOBJ/MTRL(64),NMAT(5)
      COMMON/MPROP/YPRO(2,30),YSEC(2,30),YBAC(3,30),YPHO(30),
     1ADBAC(30)
      COMMON/OBJ1/NOBJCT,NSEGO(5),CGOX(5),CGOY(5)
      COMMON/OBJ2/NNP(32),NPAT(5),MAXPAT
      COMMON/PCURT/IPTCL(5),IADEP(5),CCOND(32)
      COMMON/PHASE/NUM,NANGL,JANG,VI(25),ANG(40),NANG(25)
      COMMON/PSOLV/ISYM,IPOISS,IBPOT,RDEBY,CPOIS,NVAR,JPMAX,NBC,
     1NOMB,NODFS,NODINO,NODINB,NODINN
      COMMON/RPI/HPI,PI,HPI3,PI2,SQPI,SQPI2
      COMMON/SERCH/TA,TB,TC,V,DMA1,DMA2,DMA3,TD,TE,JP,MA,LCONV
      COMMON/BLK1/DER(3)
      COMMON/BLK2/BUF1(15),BUF2(15),BUF3(15),BUF4(15)
      COMMON/BLK3/BUF5(15),BUF6(15),BUF7(15),BUF8(20),BUF9(30)
      COMMON/BLK4/IBUF1(25),IBUF2(25),IBUF3(25),IBUF4(25)
      COMMON/BLK6/XO,YO,DH,ZM,DMB1,DMB2,DMB3,DMB4,DMB5
      COMMON/BLK7/NO,N1,N2,N3,IS,IV,IAD,JAD,IO,JO,JMM
      COMMON/BLK8/IFLA,NOB,IP,N,NSEGS,NSEGE,NPATS,NPATE
      COMMON/BLK11/CINF(22),DIN(25),DF(3),ADF(3),AUX(2,3),AUX1(6),
     1AUX2(7),RTE(3),FR(3,7)
      COMMON/BLK12/AAA(20),AAB(20),AN(20),IAS(20)
      DIMENSION NAME(3)
      DATA NAME/2HCN,2HTR,2HL /
      CALL EXEC(8,NAME)
      STOP
      END
      BLOCK DATA
C     DOUBLE PRECISION CCHB
      COMMON/CACAG/TN,TJ1,TJ2,TX,TMIN,TMAX,ITN,IT1,IT2,ITX,II,
     1JJJ,JT,JJ1
      COMMON/CACPT/IK1,IDM1,IDM2,IDM3,IDM4,DM1
      COMMON/CACP1/T1,T2,T3,T4,T5,T6,TS,LA,LB,NA,NB,NNA,NNB,
     1IU,JU,MU
      COMMON/CGUES/IK2,NK2,IFL,JGS,JGE,NF,JSF,JEF,VAL,PHIF,PHINEW,
     1CURONL,IS1
      COMMON/CHEB/CCHB(31),IORDER
      COMMON/CHFLX/I,J,JJ,J1,J2,MAXD,COEF,RATE,DEN,ID,JD,ICUT,JCUT
      COMMON/CLFLX/IK3,JK3,JM,FF0,FF1,FF2,FF3,FF4
      COMMON/CLITG/F1,F2,F3,F4,F5,TL1,TL2,TL3,TL4,V1,V2,V3,V4,V5,
     1INTYP,ICASE,ITYPE
      COMMON/CMESH/DX,DY
      COMMON/CNST1/NNEST(10),BNSTX(10,2),BNSTY(10,2)
      COMMON/CNST2/NESMAX,MAXNOD,MAXELE,NODOUT,NODIN,KELOUT,KELIN
      COMMON/CONN/NCONU,NCONA(4,48),NCONN(10,4,4)
      COMMON/CORB/NEMTP,AGEMT,BETA,VMIN,POTS,BETAP,GAMA,NPTCL,
     1LPTCL,NCHA,IORB,JORB,JONL,PINF(10),PINFI(15),TRAP(2,7)
      COMMON/CRATO/RATIO(2,8),RTPR,RTSC,RTBC,RTPH,FLXREF
      COMMON/CROOT/ACURCY,DPHI
      COMMON/CSITG/AMIN(16),AMAX(16),VL(16),VA,VB,TM1,TM2,IITYPE,
     1ILEV,ISOL,IND,NLEV
      COMMON/CSUN/SANGL,NSUNB(32),XSUNB(32),YSUNB(32),ASUNS(64),
     1NSUN
      COMMON/CURT/CSUM,CARIV,CURE,TCUR,TCURM,SUM,CURR(3)
      COMMON/CUTAG/AACUT(5,4),NEMT,NCFLG,NNCUT,CUTMIN,CUTMAX
      COMMON/CVEL/VLEV(16),NLEV,NLEVI,VLEVI(16),VLEVE(16),
     1NLEVE,RANGEE(10),COEFTE(10),POWERE(10),IDISTF(2),
     2NPARTE,NDOM(16)
      COMMON/ECONN/NCFLG(4),NFIX(8),VFIX(8),NFLOT(64),
     1NBIAS(2,8),VBIAS(8),NRES(2,8),RESTA(8),ICOND(64)
      COMMON/EMITR/NCSEC(32),NPANC(64),NPANL,NEMIT,NEMS,NEME,
     1NINTP,NSECT,NSURF
      COMMON/GENRL/LUN3,ASPOT,ICNTRL,NTOUR,ITER,
     1ACTIME,TLINIT,IOBJF,ITIME(5)
      COMMON/IODAT/LUN1,LUN2,LU,ITRACS,NTRACK(32),NITER
      COMMON/MPOBJ/MTRL(64),NMAT(5)
      COMMON/MPROP/YPRO(2,30),YSEC(2,30),YBAC(3,30),YPHO(30),
     1ADBAC(30)
      COMMON/OBJ1/NOBJCT,NSEGO(5),CGOX(5),CGOY(5)
      COMMON/OBJ2/NNP(32),NPAT(5),MAXPAT
      COMMON/PCURT/IPTCL(5),IADEP(5),CCOND(32)
      COMMON/PHASE/NUM,NANGL,JANG,VI(25),ANG(40),NANG(25)
      COMMON/PSOLV/ISYM,IPOISS,IBPOT,RDEBY,CPOIS,NVAR,JPMAX,NBC,
     1NOMB,NODFS,NODINO,NODINB,NODINN
      COMMON/RPI/HPI,PI,HPI3,PI2,SQPI,SQPI2
      COMMON/SERCH/TA,TB,TC,V,DMA1,DMA2,DMA3,TD,TE,JP,MA,LCONV
      COMMON/BLK1/DER(3)
      COMMON/BLK2/BUF1(15),BUF2(15),BUF3(15),BUF4(15)
      COMMON/BLK3/BUF5(15),BUF6(15),BUF7(15),BUF8(20),BUF9(30)
      COMMON/BLK4/IBUF1(25),IBUF2(25),IBUF3(25),IBUF4(25)
      COMMON/BLK6/XO,YO,DH,ZM,DMB1,DMB2,DMB3,DMB4,DMB5
      COMMON/BLK7/NO,N1,N2,N3,IS,IV,IAD,JAD,IO,JO,JMM
      COMMON/BLK8/IFLA,NOB,IP,N,NSEGS,NSEGE,NPATS,NPATE
      COMMON/BLK11/CINF(22),DIN(25),DF(3),ADF(3),AUX(2,3),AUX1(6),
     1AUX2(7),RTE(3),FR(3,7)
      COMMON/BLK12/AAA(20),AAB(20),AN(20),IAS(20)
      DATA IU/51/,ITRACS/0/,TLINIT/86400.0 ,ICNTRL/0/
      DATA LUN1/6/,NEMS/4/,NEME/4/
      DATA HPI,PI,HPI3,PI2,SQPI,SQPI2/1.5707963267489 ,
     A3.14159265358979 3,4.71238898038469 0,6.28318530717958 6,
     B1.77245385090551 6,1.12837916709551 3/
      DATA IORDER/31/
      END
      SUBROUTINE SORT(N)
      COMMON/BLK4/IA(25),IB(25),IBUF3(25),IBUF4(25)
C
      N1=N-1
      M=1
    1 M1=M+1
    2 IF(IA(M).LE.IA(M1)) GO TO 3
      I1=IA(M)
      I2=IB(M)
      IA(M)=IA(M1)
      IB(M)=IB(M1)
      IA(M1)=I1
      IB(M1)=I2
    3 M1=M1+1
      IF(M1.LE.N) GO TO 2
      M=M+1
      IF(M.LE.N1) GO TO 1
      RETURN
      END
      DOUBLE PRECISION FUNCTION ERFC(X)
C
C....PURPOSE..........
C
C THIS F/P COMPUTES THE COMPLEMENTARY ERROR FUNCTION USING
C    'IORDER'TH CHEBYSHEV EXPANSION.
C
C....DESCRIPTION OF THE VARIABLES..........
C
C CCHB(I):CHEBYSHEV COEFFICIENTS.
C CHB(I):CHEBYSHEV POLYNOMIALS.
C
      DOUBLE PRECISION CCHB,CHB,X,T,SUM
      COMMON/CHEB/CCHB(31),IORDER
      DIMENSION CHB(31)
C
      T=(X-3.75D0)/(X+3.75D0)
      CHB(1)=1.0D0
      CHB(2)=T
      DO 100 I=3,IORDER
      CHB(I)=2.0D0*T*CHB(I-1)-CHB(I-2)
  100 CONTINUE
      SUM=0.0D0
      DO 150 I=1,IORDER
      SUM=SUM+CHB(I)*CCHB(I)
  150 CONTINUE
      ERFC=SUM/(1.0D0+2.0D0*X)*DEXP(-X*X)
      RETURN
      END
      FUNCTION ATTR(XO,N)
C
C....PURPOSE..........
C
C THIS F/P COMPUTES THE FOLLOWING
C  ATTR(X,1)=F(X)                      FOR N=1
C  ATTR(X,2)=F(X**2)*SQRT(PI)/2.0      FOR N=2
C    WHERE F(X)=2.0/SQRT(PI)*SQRT(X)*EXP(-X)+ERFC(SQRT(X)).
C
      DOUBLE PRECISION ERFC,X,Y,SUM,SQPI2,HSQPI
      DATA SQPI2,HSQPI/1.1283791670955130D0,8.862269254527580D-1/
C
      GO TO (10,20),N
   10 IF(XO.EQ.0.0) GO TO 40
      IF(XO.GT.36.0) GO TO 60
      X=DBLE(XO)
      Y=DSQRT(X)
      SUM=SQPI2*Y*DEXP(-X)+ERFC(Y)
      GO TO 30
   20 IF(XO.EQ.0.0) GO TO 50
      IF(XO.GT.6.0) GO TO 60
      Y=DBLE(XO)
      X=Y*Y
      SUM=Y*DEXP(-X)+HSQPI*ERFC(Y)
   30 ATTR=SNGL(SUM)
      RETURN
   40 ATTR=1.0
      RETURN
   50 ATTR=SNGL(HSQPI)
      RETURN
   60 ATTR=0.0
      RETURN
      END
      END$

FTN4
      PROGRAM CNTRL(5,150)
C
C THIS CONTROLS 'XYCIC' RUNSTREAM.
C
      COMMON/GENRL/LUN3,ASPOT,ICNTRL,NTOUR,ITER,ACTIME,
     1TLINIT,IOBJF,ITIME(5)
      COMMON/IODAT/LUN1,LUN2,LU,ITRACS,NTRACK(32),NITER
      DIMENSION NAME1(3),NAME2(3),NAME3(3),NAME4(3),NAME5(3),
     1NAME6(3),NAME7(3),NAME8(3),JTIME(5),ATIME(5),BUF(32)
      DATA NAME1/2HDA,2HTA,2HI /,NAME2/2HQU,2HES,2HS /,
     1     NAME3/2HIS,2HPO,2HT /,NAME4/2HIG,2HUE,2HS /,
     2     NAME5/2HSO,2HLV,2HR /,NAME6/2HFI,2HEL,2HD /,
     3     NAME7/2HCF,2HLU,2HX /,NAME8/2HRO,2HOT,2H  /
      DATA LIMIT/0/
C
      IF(ICNTRL.EQ.0) GO TO 10
      IF(IERROR.EQ.1) GO TO 700
      GO TO (100,110,130,150,210,220,230,240),ICNTR
C
C READ ALL NECESSARY DATA.
   10 WRITE(LUN1,11)
   11 FORMAT(///20 [illegible] 5X, "PSYCHIC" STARTS, ,5X,20 [illegible])
      ICNTRL=1
      CALL EXEC(8,NAME1)
  100 WRITE(LUN1,101)
  101 FORMAT(5X,'FINISHED READING ALL DATA. ')
C
      WRITE(LUN1,103)
```

```
 105 FORMAT(    *****KSPOT /***** )
     READ(1,*) KSPOT
     GO TO (105,130,140,140),KSPOT
C
C  COMPUTE GUESS FIELD.
 105 ICNTRL=2
     CALL EXEC(8,NAME2)
 110 WRITE(LUN1,111)
 111 FORMAT(5X,'GUESS FIELD HAS BEEN OBTAINED.')
     GO TO 160
C
C  READ SURFACE POTENTIALS.
 130 ICNTRL=3
     CALL EXEC(8,NAME3)
 130 WRITE(LUN1,131)
 131 FORMAT(5X,'FINISHED READING SURFACE POTENTIALS.')
     GO TO 160
C
C  SET UP INITIAL GUESS FIELD.
 140 ICNTRL=4
     CALL EXEC(8,NAME4)
 150 WRITE(LUN1,151)
 151 FORMAT(5X,'FINISHED READING INITIAL GUESS.')
     GO TO 170
C
C*************************************
C  MAJOR ITERATION STARTS.
C
 160 ITER=0
     NTOUR=0
 170 IOBJF=0
     ACTIME=0.0
     WRITE(LUN1,171)
 171 FORMAT(//20'*',5X,'MAJOR ITERATION STARTS.',5X,20'*'//)
C
     GO TO (200,200,205,220),KSPOT
 200 CALL EXEC(11,ITIME)
     NTOUR=NTOUR+1
     ITER=ITER+1
C
C  CALL POISSON SOLVER.
 205 ICNTRL=5
     CALL EXEC(8,NAME5)
C
C  COMPUTE POTENTIAL DERIVATIVES.
 210 ICNTRL=6
     CALL EXEC(8,NAME6)
C
C  COMPUTE CURRENT AT 'PEP' AND ORGANIZE TOTAL CURRENT FOR SECTOR.
 220 ICNTRL=7
     CALL EXEC(8,NAME7,0)
C
C  COMPUTE OBJECTIVE FUNCTION AND RESET NECESSARY PARAMETERS FOR
C   NEXT ITERATION.
 230 ICNTRL=8
     CALL EXEC(8,NAME8)
C
C  CHECK CPU TIME PER ITERATION AND ACCUMULATED TIME(IN SECONDS).
 240 CALL EXEC(11,JTIME)
     DO 500 I=1,5
     ATIME(I)=JTIME(I)-ITIME(I)
 500 CONTINUE
     CPUT=ATIME(2)/60.0+ATIME(3)+60.0*ATIME(4)+1440.0*ATIME(5)
     ACTIME=ACTIME+CPUT
     ITRAC=NTRACK(21)
     ISECT=(ITER-1)/32
     II=ITER-ISECT*32
     CALL EXEC(1,LU,BUF,64,ITRAC,ISECT+32)
     BUF(II)=CPUT
     CALL EXEC(2,LU,BUF,64,ITRAC,ISECT+32)
     CALL EXEC(1,LU,BUF,64,ITRAC,ISECT+64)
     BUF(II)=ACTIME
     CALL EXEC(2,LU,BUF,64,ITRAC,ISECT+64)
     WRITE(LUN1,1010) ITER,ACTIME,CPUT
1010 FORMAT(5X,'# OF ITERATIONS=',I3,5X,'ACCUMULATED TIME=',
    1F10.1,' MIN',5X,'CPU TIME=',F10.1,' MIN')
     IF(IOBJF.EQ.1) GO TO 510
C  IF ACCUMULATED TIME HAS EXCEEDED THE TIME LIMIT, STOP PROGRAM.
     IF(LIMIT.EQ.1.AND.ACTIME.GT.TLIMIT) GO TO 600
     GO TO 200
C
C  MAJOR ITERATION ENDS.
C*****************************************************************
C
 510 WRITE(LUN1,1020)
1020 FORMAT(5X,'**FINAL CONVERGENCE HAS BEEN OBTAINED.**')
     STOP 0001
 600 WRITE(LUN1,1030)
1030 FORMAT(5X,'**ACCUMULATED TIME HAS EXCEEDED THE TIME'
    1' LIMIT.**')
     STOP 0002
 700 WRITE(LUN1,1040)
1040 FORMAT(5X,'**ERROR IS DETECTED....STOP RUNNING THIS'
    1' JOB.**')
     STOP 0003
     END
     END$



FTN4
$FMA(AEMA3,0)
      PROGRAM SOLVR(5,150)
C
C THIS S/P SETS UP VARIABLES WHICH ARE NEEDED TO CALCULATE
C  POTENTIAL.
C
      COMMON/AEMA3/IB(2560,4),IWORK(256),POT(2560),AY(2560),
     1EY(2560),AM(5840),RWORK(256),CION(128),CELE(128),CPRO(128),
     2CPRON(128),CSEC(128),CSECN(128),CBSC(128),CBSCN(128),
     3CPHO(128),CPHON(128),PSECT(128),CSECT(128),PEMIT(128),
     4XSP(128),YSP(128),XEP(128),YEP(128),XSSEG(64),YSSEG(64),
     5XESEG(64),YESEG(64),SLPSEG(64),IROOT(128),IOK(128),
     6NOBEMT(128),NCONU(128),NCONV(128),NCONW(128),NCONX(128),
     7NEMIS(128),NEMIE(128),NSECT(128),NCOND(128),NPPAN(64,2),
     8NSUMP(128),ISLP(64),IVEC(64),NPAT(64),NCP(128),NCHA(128),
     9NCNB(384),NCNC(128),NSP(128),NEP(128),NSURF(128)
      COMMON/CNST2/NESMAX,MAXNOD,MAXELE,NODOUT,NODIN,KFLOUT,KELIN
      COMMON/IODAT/LUN1,LUN2,LU,ITRACS,NTRACK(32),MITER
      COMMON/PSOLV/ISYM,IPOISS,IBPOT,RDEDY,CPOIS,NVAR,JPMAX,NBC,
     1NOND,NODES,NODINO,NODINS,NODINN
      DIMENSION IBC(256),JBC(256),YBC(256)
      DIMENSION JA(5120),IA(512),NC(512,8),LBUF(64)
      DIMENSION NAME1(3),NAME2(3),BUFR(256),IBUFR(256),JBUFR(256)
      EQUIVALENCE (IBUFR,LBUF),(IROOT,IA)
      EQUIVALENCE (IWORK,IBC),(IB,JA,JBC),(IB(1,3),NC),(EY,YBC)
      DATA NAME1/2HSO,2HIS,2H1 /,NAME2/2HIC,2HCG,2H /
C
      WRITE(LUN1,999)
 999 FORMAT(20'*',5X,'SOLVR STARTS',5X,20'*')
C
```

```
      IPOISS=1
      NSTOP=NVAR/256+1
      ISECT=0
      ISTART=0
      DO 40 N=1,NSTOP
      CALL EXEC(1,LU,BUFR,512,NTRACK(8),ISECT)
      DO 50 I=1,256
      AY(ISTART+I)=BUFR(I)
   50 CONTINUE
      ISECT=ISECT+8
      ISTART=ISTART+256
   40 CONTINUE
C     WRITE(LUN1,753) (AY(I),I=1,NVAR)
C 753 FORMAT('*'/(1P10E13.5))
      NSTOP=NBC/256+1
      ISECT=0
      ISTART=0
      DO 60 N=1,NSTOP
      CALL EXEC(1,LU,IBUFR,256,NTRACK(9),ISECT)
      CALL EXEC(1,LU,JBUFR,256,NTRACK(9),ISECT+24)
      DO 70 I=1,256
      II=ISTART+I
      IBC(II)=IBUFR(I)
      JBC(II)=JBUFR(I)
   70 CONTINUE
      ISTART=ISTART+256
      ISECT=ISECT+4
   60 CONTINUE
      ISECT=48
      ISTART=0
      DO 80 N=1,NSTOP
      CALL EXEC(1,LU,BUFR,512,NTRACK(9),ISECT)
      DO 90 I=1,256
      YBC(ISTART+I)=BUFR(I)
   90 CONTINUE
      ISTART=ISTART+256
      ISECT=ISECT+8
   80 CONTINUE
C     WRITE(LUN1,738) NBC
C 738 FORMAT('NBC=',I5)
C     WRITE(LUN1,739) (IBC(I),I=1,NBC)
C 739 FORMAT('IBC'/(25I5))
C     WRITE(LUN1,740) (JBC(I),I=1,NBC)
C 740 FORMAT('JBC'/(25I5))
      DO 100 I=1,NBC
      N=IBC(I)+1-NODINS
C     N=IBC(I)
      NODE=JBC(I)
      AY(N)=AY(N)-YBC(I)*POT(NODE)
  100 CONTINUE
C     WRITE(LUN1,753) (AY(I),I=1,NVAR)
      NSTOP=MAXNOD/256+1
      ISTART=0
      ISECT=0
      DO 110 N=1,NSTOP
      DO 120 I=1,256
      BUFR(I)=POT(I+ISTART)
  120 CONTINUE
      CALL EXEC(2,LU,BUFR,512,NTRACK(11),ISECT)
      ISTART=ISTART+256
      ISECT=ISECT+8
  110 CONTINUE
C
C     GO TO(200,300),IPOISS
C  FOR SOR-METHOD.
C 200 CALL EXEC(8,NAME1)
C
C  FOR ICCG-METHOD.
  300 NSTOP=(NVAR+1)/256+1
      ISECT=0
      ISTART=0
      DO 310 N=1,NSTOP
      CALL EXEC(1,LU,IBUFR,256,NTRACK(6),ISECT)
      DO 320 I=1,256
      IA(ISTART+I)=IBUFR(I)
  320 CONTINUE
      DO 340 I=1,256
      BUFR(I)=AY(ISTART+I)
  340 CONTINUE
      CALL EXEC(2,LU,BUFR,512,NTRACK(7),ISECT*2)
      ISECT=ISECT+4
      ISTART=ISTART+256
  310 CONTINUE
      NSTOP=JPMAX/256+1
      ITRACK=NTRACK(5)
      ISECT=0
      ISTART=0
      DO 350 N=1,NSTOP
      CALL EXEC(1,LU,IBUFR,256,ITRACK,ISECT)
      DO 360 I=1,256
      JA(ISTART+I)=IBUFR(I)
  360 CONTINUE
      ISECT=ISECT+4
      ISTART=ISTART+256
      IF(MOD(N,24).NE.0) GO TO 350
      ITRACK=ITRACK+1
      ISECT=0
  350 CONTINUE
C     WRITE(LUN1,753) (POT(I),I=1,NVAR)
C     WRITE(LUN1,753) (AP(I),I=1,NVAR)
C     WRITE(LUN1,888) (JA(I),I=1,JPMAX)
C 888 FORMAT('**JA**'/(25I5))
C     WRITE(LUN1,879) (IA(I),I=1,NVAR)
C 879 FORMAT('**IA**'/(25I5))
      IF(IPOISS.EQ.2) CALL EXEC(8,NAME2)
      ITRACK=NTRACK(3)
      ISECT=0
      ISTART=0
      DO 400 N=1,NSTOP
      CALL EXEC(1,LU,BUFR,512,ITRACK,ISECT)
      DO 410 I=1,256
      AM(ISTART+I)=BUFR(I)
  410 CONTINUE
      ISECT=ISECT+8
      ISTART=ISTART+256
      IF(MOD(N,12).NE.0) GO TO 400
      ITRACK=ITRACK+1
      ISECT=0
  400 CONTINUE
C
      NSTOP=NODINN/8+1
      ISECT=0
      ISTART=0
      DO 420 N=1,NSTOP
      CALL EXEC(1,LU,LBUF,64,ITRACS+5,ISECT)
      M=0
      DO 430 I=1,8
      II=ISTART+I
      DO 430 J=1,8
      M=M+1
      NC(II,J)=LBUF(M)
  430 CONTINUE
```

```
      ISECT=ISECT+1
      ISTART=ISTART+1
  170 CONTINUE
C
      MMM=0
      IF(MMM.EQ.0) GO TO 450
      DO 1000 N=1,NVAR
      WRITE(LUN1,887) N
  887 FORMAT(' N=',I4)
      I1=IA(N)
      I2=IA(N+1)-1
      WRITE(LUN1,888) (JA(I),I=I1,I2)
      WRITE(LUN1,889) (AM(I),I=I1,I2)
  888 FORMAT(12I10)
  889 FORMAT(1P12E10.2)
 1000 CONTINUE
      WRITE(LUN1,984)
  984 FORMAT(' ** NPOIS',7X,'E      N      W      S      NE      NW',
     1'      SW      SE')
      DO 440 I=1,NODINN
      N=NODINO+I-1
      WRITE(LUN1,987) N,(NC(I,J),J=1,8)
  987 FORMAT(' N=',I4,5X,8I6)
  440 CONTINUE
  450 CALL EXEC(8,NAME1)
C
      STOP
      END
      END$
```

```
FTN4
$EMA(AEMA3,0)
      PROGRAM POIS1(5,150)
C
C.....PURPOSE.............
C
C THIS S/P SOLVES POISSON'S EQUATION
C       div(grad(PHI))+C*(PHI)=0.
C             WHERE "C" IS CONSTANT, AND "PHI" IS POTENTIAL.
C
      COMMON/AEMA3/IG(2560,4),IWORK(256),POT(2560),AY(2560),
     1EY(2560),AM(5040),RWORK(256),CION(128),CELE(128),CPRO(128),
     2CPROM(128),CSEC(128),CSECM(128),CBSC(128),CBSCM(128),
     3CPHO(128),CPHOM(128),PSECT(128),CSECT(128),PEMIT(128),
     4XSP(128),YSP(128),XEP(128),YEP(128),XSSEG(64),YSSEG(64),
     5XESEG(64),YESEG(64),SLPSEG(64),IROOT(128),IOK(128),
     6NODEMT(128),NCONU(128),NCONV(128),NCONW(128),NCONX(128),
     7NEMTS(128),NEMTE(128),NSECT(128),NCOND(128),NPPAN(64,2),
     8NSUMP(128),ISLP(64),IVEC(64),NPAT(64),NCP(128),NCNA(128),
     9NCNB(384),NCNC(128),NSP(128),NEP(128),NSURF(128)
      COMMON/CNST1/NNEST(10),BNSTX(10,2),BNSTY(10,2)
      COMMON/CNST2/NESMAX,MAXNOD,MAXELE,NODOUT,NODIN,KELOUT,KELIN
      COMMON/EMITR/NCSEC(32),NPANC(64),NPANL,NEMIT,NEMS,NEME,
     1NINTP,MSECT,MSURF
      COMMON/IODAT/LUN1,LUN2,LU,ITRACS,NTRACK(32),MITER
      COMMON/PSOLV/ISYM,IPOISS,IBPOT,RDEBY,CPOIS,NVAR,JPMAX,NBC,
     1NOMB,NODFS,NODINO,NODINS,NODINN
      DIMENSION JA(5120),IA(512),NC(512,8),NAME(3)
      DIMENSION IBUF(64),NNI(10),NNJ(10)
      DIMENSION A(24),BOFF(2,32),BDIA(2,32),IKMAX(4)
      EQUIVALENCE (IG,JA),(IG(1,3),NC),(IROOT,IA)
      DATA EPSMAX/1.0E-3/,ITMAX/2000/
      DATA NAME/2HWE,2HMA,2H4 /
C
      WRITE(LUN1,999)
  999 FORMAT(20'.',5X,'"POIS1" STARTS',5X,20'.')
      A1=3.141592654/SQRT(FLOAT(MAXNOD))
      OMEGA=2.0*(1.0-SIN(A1))/COS(A1)**2
      CALL EXEC(1,LU,BDIA,128,ITRACS+4,81)
      CALL EXEC(1,LU,BOFF,128,ITRACS+4,83)
      CALL EXEC(1,LU,IBUF,64,ITRACS+2,50)
      DO 30 I=1,NESMAX
      NNI(I)=IBUF(I)
      NNJ(I)=IBUF(I+10)
   30 CONTINUE
      PMAX=0.0
      DO 50 N=1,NSURF
      NODE=NSURF(N)
      PPOT=ABS(POT(NODE))
      IF(PPOT.GT.PMAX) PMAX=PPOT
   50 CONTINUE
      REPS=EPSMAX
      IF(PMAX.GT.1.0E-3) REPS=EPSMAX*PMAX
      INES=NESMAX-1
C
      ITER=0
 2000 ITER=ITER+1
      EPS=0.0
      N=0
      DO 100 I=1,NESMAX
      II=NESMAX+1-I
      NI=NNI(II)
      NJ=NNJ(II)
      JMAX=NNEST(II)
      JCOUNT=1
      DO 110 K=1,2
      IKMAX(K*2-1)=NI-1
      IKMAX(K*2)=NJ-1
  110 CONTINUE
      C=CPOIS*(DX*FLOAT(II))**2/3.0
      C1=(1.0-C)*72.0
      C2=(3.0-4.0*C)*24.0
      C3=(2.0-C)*64.0
      A(1)=(1.0+C)*8.0/C1
      A(2)=(5.0+26.0*C)/C1
      A(3)=(5.0+2.0*C)/C1
      A(4)=(3.0+2.0*C)*6.0/C1
      A(5)=(4.0+C)*2.0/C1
      A(6)=(1.0+2.0*C)*6.0/C2
      A(7)=(3.0+2.0*C)*6.0/C2
      A(8)=(1.0+C)*8.0/C2
      A(9)=(4.0+C)*2.0/C2
      A(10)=(5.0+2.0*C)/C2
      A(11)=(1.0+10.0*C)*2.0/C2
      A(12)=(4.0+C)*4.0/C3
      A(13)=(1.0+C)*16.0/C3
      A(14)=(10.0+12.0*C)/C3
      A(15)=(2.0+C)*6.0/C3
      A(16)=(10.0+C)/C3
      A(17)=(2.0+5.0*C)/C3
      IF(II.LT.NESMAX) GO TO 170
C
C  FOR THE OUTERMOST BOUNDARIES.
C
      LO=2
      M=(NI+NJ)*2
      A(18)=(1.0+C)/6.0
      A(19)=(1.0+C)/3.0
      A(20)=(4.0+C)/3.0
      A(21)=(5.0-C)*2.0/3.0
      A(22)=(2.0-C)/3.0
      DO 120 J=1,4
      KMAX=IKMAX(J)
      N=N+1
      IF(J.GE.3) GO TO 122
      IU=1
      L1=1
      L2=1
      IW=0
      GO TO 125
  122 IU=-1
      L1=J-2
      L2=KMAX+1
      IW=1
      GO TO 130
  125 IF(J.NE.1) GO TO 130
      KSTART=2
      SUM=((A(18)-BOFF(1,1))*POT(N+1)+(A(18)-BOFF(2,1))*
     1POT(M)+A(20)*POT(M+1))/(A(22)+BDIA(1,1))
      GO TO 140
  130 KSTART=3
      L3=3-L1
      L4=1
      IF(J.EQ.2.OR.J.EQ.3) L4=IKMAX(J-1)+1
      SUM=((A(18)-BOFF(L1,L2))*POT(N+1)+(A(18)-BOFF(L3,L4))*
     1POT(M-1)+A(20)*POT(M))/(A(22)+BDIA(L1,L2+IW))
      RES=SUM-POT(N)
      POT(N)=POT(N)+OMEGA*RES
      EPS=EPS+ABS(RES)
      N=N+1
      L2=L2+IU
      SUM=((A(18)-BOFF(L1,L2-IU))*POT(N-1)+(A(18)-BOFF(L1,L2))*
     1POT(N+1)+A(19)*POT(M)+A(20)*(POT(M+1)+POT(N-2)))/
     2(A(21)+BDIA(L1,L2+IW))
  140 RES=SUM-POT(N)
      POT(N)=POT(N)+OMEGA*RES
      EPS=EPS+ABS(RES)
      DO 150 K=KSTART,KMAX
      N=N+1
      M=M+1
      L2=L2+IU
      MS=M+1
      IF(J.EQ.4.AND.K.EQ.KMAX) MS=N+2
      SUM=((A(18)-BOFF(L1,L2-IU))*POT(N-1)+(A(18)-BOFF(L1,L2))*
     1POT(N+1)+A(19)*POT(M)+A(20)*(POT(MS)+POT(M-1)))/
     2(A(21)+BDIA(L1,L2+IW))
      RES=SUM-POT(N)
      POT(N)=POT(N)+OMEGA*RES
      EPS=EPS+ABS(RES)
  150 CONTINUE
      N=N+1
      M=M+1
      L2=L2+IU
      IF(J.EQ.4) GO TO 155
      SUM=((A(18)-BOFF(L1,L2-IU))*POT(N-1)+(A(18)-BOFF(L1,L2))*
     1POT(N+1)+A(19)*POT(M)+A(20)*(POT(N+2)+POT(M-1)))/
     2(A(21)+BDIA(L1,L2+IW))
      GO TO 160
  155 SUM=((A(18)-BOFF(L1,L2-IU))*POT(N-1)+(A(18)-BOFF(L1,L2))*
     1POT(LO-1)+A(19)*POT(N+1)+A(20)*(POT(LO)+POT(M-1)))/
     2(A(21)+BDIA(L1,L2+IW))
  160 RES=SUM-POT(N)
      POT(N)=POT(N)+OMEGA*RES
      EPS=EPS+ABS(RES)
  120 CONTINUE
      GO TO 230
C
C  FOR THE OUTER EDGE OF GRID.
C
  170 CONTINUE
      M=N+(NI+NJ)*2
      L=LO
      LO=N+2
      DO 180 J=1,4
      KMAX=IKMAX(J)-1
      N=N+1
      IF(J.NE.1) GO TO 190
      SUM=A(1)*POT(L-1)+A(2)*(POT(N-1)+POT(L))+A(3)*(POT(N-2)
     1+POT(L+1))+A(4)*(POT(N+1)+POT(M))+A(5)*POT(M+1)
      GO TO 200
  190 L=L+3
      SUM=A(1)*POT(L-1)+A(2)*(POT(L-2)+POT(L))+A(3)*(POT(L-3)
     1+POT(L+1))+A(4)*(POT(N+1)+POT(N-1))+A(5)*POT(M)
      RES=SUM-POT(N)
      POT(N)=POT(N)+OMEGA*RES
      EPS=EPS+ABS(RES)
      N=N+1
      SUM=A(6)*(POT(L)+POT(L+1))+A(7)*(POT(N+1)+POT(N-1))+
     1A(8)*POT(M)+A(9)*(POT(M+1)+POT(M-2))
  200 RES=SUM-POT(N)
      POT(N)=POT(N)+OMEGA*RES
      EPS=EPS+ABS(RES)
      DO 210 K=2,KMAX,2
      IF(J.GE.2.AND.K.EQ.2) GO TO 205
      N=N+1
      M=M+1
      SUM=A(6)*(POT(L)+POT(L+1))+A(7)*(POT(N+1)+POT(N-1))+
     1A(8)*POT(M)+A(9)*(POT(M+1)+POT(M-1))
      RES=SUM-POT(N)
      POT(N)=POT(N)+OMEGA*RES
      EPS=EPS+ABS(RES)
  205 N=N+1
      M=M+1
      L=L+1
      MS=M+1
      IF(J.EQ.4.AND.K.EQ.KMAX) MS=N+2
      SUM=A(10)*(POT(L+1)+POT(L-1))+A(11)*POT(L)+A(7)*
     1(POT(N+1)+POT(N-1))+A(8)*POT(M)+A(9)*(POT(MS)+POT(M-1))
      RES=SUM-POT(N)
      POT(N)=POT(N)+OMEGA*RES
      EPS=EPS+ABS(RES)
  210 CONTINUE
      N=N+1
      M=M+1
      IF(J.EQ.4) GO TO 215
      SUM=A(6)*(POT(L)+POT(L+1))+A(7)*(POT(N+1)+POT(N-1))+
     1A(8)*POT(M)+A(9)*(POT(N+2)+POT(M-1))
      GO TO 220
  215 SUM=A(6)*(POT(L)+POT(L+1))+A(7)*(POT(L+2)+POT(N-1))+
     1A(8)*POT(N+1)+A(9)*(POT(L+3)+POT(M-1))
  220 RES=SUM-POT(N)
      POT(N)=POT(N)+OMEGA*RES
      EPS=EPS+ABS(RES)
  180 CONTINUE
C
  230 CONTINUE
      NI=NI-2
      NJ=NJ-2
      M=N+(NI+NJ)*2
      L=LO
      LO=N+2
```

```
      DO 240 K=1,2
      IKMAX(K*2-1)=N1-1
      IKMAX(K*2)=N-1
  40  CONTINUE
      ICOUNT=ICOUNT+1
      IF(ICOUNT.LE.LMAX) GO TO 300
C
C  FOR THE INTERIOR SPACE OF GRID.
C
      DO 250 J=1,4
      KMAX=IKMAX(J)
      N=N+1
      IF(J.NE.1) GO TO 260
      KSTART=2
      SUM=A(12)*(POT(L+1)+POT(L-1)+POT(M+1)+POT(N-2))+
     1A(13)*(POT(L)+POT(M)+POT(N+1)+POT(N-1))
      GO TO 270
  260 KSTART=3
      L=L+3
      SUM=A(12)*(POT(L+1)+POT(L-1)+POT(M)+POT(L-3))+
     1A(13)*(POT(L)+POT(L-2)+POT(N+1)+POT(N-1))
      RES=SUM-POT(N)
      POT(N)=POT(N)+OMEGA*RES
      EPS=EPS+ABS(RES)
      N=N+1
      L=L+1
      SUM=A(12)*(POT(L+1)+POT(L-1)+POT(M+1)+POT(N-2))+
     1A(13)*(POT(L)+POT(M)+POT(N+1)+POT(N-1))
  270 RES=SUM-POT(N)
      POT(N)=POT(N)+OMEGA*RES
      EPS=EPS+ABS(RES)
      DO 280 K=KSTART,KMAX
      N=N+1
      M=M+1
      L=L+1
      MS=M+1
      IF(J.EQ.4.AND.K.EQ.KMAX) MS=M+2
      SUM=A(12)*(POT(L+1)+POT(L-1)+POT(MS)+POT(M-1))+
     1A(13)*(POT(L)+POT(M)+POT(N+1)+POT(N-1))
      RES=SUM-POT(N)
      POT(N)=POT(N)+OMEGA*RES
      EPS=EPS+ABS(RES)
  280 CONTINUE
      N=N+1
      M=M+1
      L=L+1
      IF(J.EQ.4) GO TO 285
      SUM=A(12)*(POT(L+1)+POT(L-1)+POT(M+2)+POT(M-1))+
     1A(13)*(POT(L)+POT(M)+POT(N+1)+POT(N-1))
      GO TO 290
  285 SUM=A(12)*(POT(L+1)+POT(L-1)+POT(L+3)+POT(M-1))+
     1A(13)*(POT(L)+POT(N+1)+POT(L+2)+POT(N-1))
  290 RES=SUM-POT(N)
      POT(N)=POT(N)+OMEGA*RES
      EPS=EPS+ABS(RES)
  250 CONTINUE
      GO TO 230
C
C  FOR THE INNER EDGE OF GRID.
C
  300 CONTINUE
      IF(II.EQ.1) GO TO 345
      DO 310 J=1,4
      KMAX=IKMAX(J)
      N=N+1
      IF(J.NE.1) GO TO 320
      SUM=A(12)*(POT(L+1)+POT(L-1)+POT(N+1)+POT(N-2))+A(13)*
     1(POT(L)+POT(M)+POT(N+1)+POT(N-1))
      RES=SUM-POT(N)
      POT(N)=POT(N)+OMEGA*RES
      EPS=EPS+ABS(RES)
      N=N+1
      M=M+1
      L=L+1
      SUM=A(12)*(POT(L+1)+POT(L-1)+POT(M-1))+A(13)*
     1(POT(L)+POT(N+1)+POT(N-1))+A(14)*POT(M)+A(15)*POT(M+1)+
     2A(16)*POT(M+2)
      GO TO 330
  320 L=L+3
      SUM=A(12)*(POT(L+1)+POT(L-1)+POT(M)+POT(L-3))+A(13)*
     1(POT(L)+POT(L-2)+POT(N+1)+POT(N-1))
      RES=SUM-POT(N)
      POT(N)=POT(N)+OMEGA*RES
      EPS=EPS+ABS(RES)
      N=N+1
      L=L+1
      SUM=A(12)*(POT(L+1)+POT(L-1)+POT(M-2))+A(13)*
     1(POT(L)+POT(N+1)+POT(N-1))+A(14)*POT(M)+A(15)*POT(M+1)+
     2A(16)*POT(M+2)
  330 RES=SUM-POT(N)
      POT(N)=POT(N)+OMEGA*RES
      EPS=EPS+ABS(RES)
      DO 340 K=3,KMAX
      N=N+1
      M=M+2
      L=L+1
      MS=M+2
      IF(J.EQ.4.AND.K.EQ.KMAX) MS=M+2
      SUM=A(12)*(POT(L+1)+POT(L-1))+A(13)*
     1(POT(L)+POT(N+1)+POT(N-1))+A(17)*POT(M)+A(15)*(POT(M+1)+
     2POT(M-1))+A(16)*(POT(MS)+POT(M-2))
      RES=SUM-POT(N)
      POT(N)=POT(N)+OMEGA*RES
      EPS=EPS+ABS(RES)
  340 CONTINUE
      N=N+1
      M=M+2
      L=L+1
      IF(J.EQ.4) GO TO 342
      SUM=A(12)*(POT(L+1)+POT(L-1)+POT(M+2))+A(13)*
     1(POT(L)+POT(N+1)+POT(N-1))+A(14)*POT(M)+A(15)*POT(M-1)+
     2A(16)*POT(M-2)
      GO TO 344
  342 SUM=A(12)*(POT(L+1)+POT(L-1)+POT(L+3))+A(13)*
     1(POT(L)+POT(L+2)+POT(N-1))+A(14)*POT(M+1)+
     2A(15)*POT(M-1)+A(16)*POT(M-2)
  344 RES=SUM-POT(N)
      POT(N)=POT(N)+OMEGA*RES
      EPS=EPS+ABS(RES)
  310 CONTINUE
  100 CONTINUE
C
C  FOR THE INNERMOST GRID.
C
  345 CONTINUE
      MAX=NODINS-1
      I=0
      DO 350 N=NODINO,MAX
      I=I+1
      SUM1=0.0
      SUM2=0.0
      SUM1=SUM1+POT(M(I+1))
      SUM2=SUM2+POT(M(I+4))
  360 CONTINUE
      SUM=A(12)*SUM1+A(11)*SUM2
      RES=SUM-POT(N)
      POT(N)=POT(N)+OMEGA*RES
      EPS=EPS+ABS(RES)
  350 CONTINUE
C
      DO 370 N=1,NVAR
      IO=IA(N)
      I1=IO+1
      I2=IA(N+1)-1
      NO=NODINS-1+N
      SUM=0.0
      DO 380 I=I1,I2
      SUM=SUM+AM(I)*POT(IA(I))
  380 CONTINUE
      SUM=(AY(N)-SUM)/AM(IO)
      RES=SUM-POT(NO)
      POT(NO)=SUM
      EPS=EPS+ABS(RES)
  370 CONTINUE
C
C     WRITE(LUN1,988) ITER,EPS
C 988 FORMAT(' ITER=',I3,5X,' EPS=',1PE13.5)
      IF(ITER.GE.ITMAX) GO TO 510
      IF(EPS.LT.REPS) GO TO 510
      GO TO 2000
  500 CALL EXEC(8,NAME,10,ITER)
C
  510 CONTINUE
      CALL EXEC(8,NAME,20,ITER)
      STOP
      END
      END$
```

```
FTN4
$EMA(AEMA3,0)
      PROGRAM WEMA4(5,150)
      COMMON/AEMA3/IG(2560,4),IWORK(256),POT(2560),AP(2560),
     1EY(2560),AM(5840),RWORK(256),CION(128),CELE(128),CPRO(128),
     2CPROM(128),CSEC(128),CSECM(128),CBSC(128),CBSCM(128),
     3CPMO(128),CPMOM(128),PSECT(128),CSECT(128),PERIT(128),
     4XSP(128),YSP(128),XEP(128),YEP(128),XSSEG(64),YSSEG(64),
     5XESEG(64),YESEG(64),SLPSEG(64),IROOT(128),IOK(128),
     6NODENT(128),NCONU(128),NCONV(128),NCONW(128),NCONX(128),
     7NEMTS(128),NEMTE(128),NSECT(128),NCOND(128),NPPAN(64,2),
     8NSUNP(128),ISLP(64),IVEC(64),NPAT(64),NCP(128),NCHA(128),
     9NCNB(384),NCNC(128),NSP(128),NEP(128),NSURF(128)
      COMMON/CNST2/NESMAX,MAXNOD,MAXELE,NODOUT,NODIN,KELOUT,KELIN
      COMMON/IODAT/LUN1,LUN2,LU,ITRACS,NTRACK(32),NITER
      COMMON/PSOLV/ISYM,IPOISS,IBPOT,RDEBY,CPOIS,NVAR,JPMAX,NBC,
     1NOMB,NODFS,NODINO,NODINS,NODINN
      DIMENSION XO(2560),YO(2560)
      DIMENSION BUFR(256),IBUFR(128)
      DIMENSION NAME(3),LIST(5)
      EQUIVALENCE (AM(1),XO),(AM(2561),YO)
      DATA NAME/2HDA,2HTA,2H3 /
C
      CALL RMPAR(LIST)
      WRITE(LUN1,999)
  999 FORMAT(20'.',5X,'"WEMA4" STARTS',5X,20'.')
      ICODE=LIST(1)
      CALL EXEC(1,LU,IBUFR,128,NTRACK(19),16)
      IBUFR(NITER)=LIST(2)
      CALL EXEC(2,LU,IBUFR,128,NTRACK(19),16)
      NSTOP=MAXNOD/256+1
      ISTART=0
      ISECT=0
      NTR=NTRACK(ICODE)
      DO 100 N=1,NSTOP
      DO 110 I=1,256
      BUFR(I)=POT(I+ISTART)
  110 CONTINUE
      CALL EXEC(2,LU,BUFR,512,NTR,ISECT)
      ISTART=ISTART+256
      ISECT=ISECT+8
  100 CONTINUE
      IF(ICODE.EQ.10) GO TO 300
C     WRITE(LUN1,151) (POT(I),I=1,MAXNOD)
C 151 FORMAT(1P10E13.5)
      NSTOP=MAXNOD/256+1
      ISTART=0
      ISECT=0
      DO 160 N=1,NSTOP
      DO 170 I=1,256
      BUFR(I)=POT(I+ISTART)
  170 CONTINUE
      CALL EXEC(2,LU,BUFR,512,NTRACK(11),ISECT)
      CALL EXEC(1,LU,BUFR,512,NTRACK(14),ISECT)
      DO 180 I=1,256
      XO(I+ISTART)=BUFR(I)
  180 CONTINUE
      CALL EXEC(1,LU,BUFR,512,NTRACK(15),ISECT)
      DO 190 I=1,256
      YO(I+ISTART)=BUFR(I)
  190 CONTINUE
      ISTART=ISTART+256
      ISECT=ISECT+8
  160 CONTINUE
      NSTOP=MAXELE/128+1
      DO 200 J=1,2
      ISTART=0
      ISECT=0
      IF(J.EQ.2) ISECT=40
      DO 210 N=1,NSTOP
      CALL EXEC(1,LU,IBUFR,128,NTRACK(16),ISECT)
      DO 220 I=1,128
      IG(I+ISTART,J)=IBUFR(I)
  220 CONTINUE
      ISECT=ISECT+2
      ISTART=ISTART+128
  210 CONTINUE
  200 CONTINUE
      DO 230 J=3,4
      ISTART=0
      ISECT=0
      IF(J.EQ.4) ISECT=40
      DO 240 N=1,NSTOP
[illegible]
[illegible]
[illegible]
  250 CONTINUE
[illegible]
      ISTART=ISTART+128
  240 CONTINUE
[illegible]
```

```
      WRITE(CONT,501) IT,IT
    1 FORMAT(' CONVERGENCE HAS NOT BEEN REACHED AFTER ',I3,
     1 ' ITERATIONS ')
      IERROR=1
      CALL EXEC(6,NAME)
      STOP
      END
      END$

 FN4
 ALMA4,ALMA4,0)
      PROGRAM PLOT1(5,150)
C
C THIS P GENERATES POTENTIAL CONTOUR PLOTS. THE SATELLITE
C   PERIMETER IS ALSO PLOTTED WITH THE CONTOURS.
C
      COMMON/ALMA4/POT(2560),XG(2560),YG(2560),IG(2560,4),
     1ZP(50,50),XP(50),YP(50),IBUFP(50),IL(50,50)
      COMMON/BLK1/NNI(10),NNJ(10),KELOUT,KMAX,DX,DY
      COMMON/BLK8/NOUTER(10),NNEST(10),S(4,4),NCONN(10,4,4),
     1NCONA(130),IGA(4)
      COMMON/BLK15/BUF7(1,1),IBUF1(25),IBUF2(25)
      COMMON/FILE2/IMAX,MB,II,NESMAX,MAXNOD,MAXELE,NODOUT
      COMMON/PLOT1/IPLOT(1),MAX,IMAX1,JMAX1,NS,NA,ISYM,IFLAG
      DIMENSION XPP(4),YPP(4),CNP(30)
C
      IDIMX=57
      IEND=NESMAX-3
C
      MM=2
      MMM=1
C
      ZM=-1.0E+28
      SEP=-0.1
      IF(ISYM.EQ.1) GO TO 10
      AAX=6.5
      AAY=5.5
      SC=1.6
      GO TO 20
   10 AAX=2.0
      AAY=1.0
      SC=0.5
   20 SX=SC*DX
      SY=SC*DY
      SCZ=1.0
C
      CALL PAPLT(NCN,CNP)
C
      IF(MM.EQ.1) GO TO 47
      CALL START
      CALL SYMBOL(0.0,0.0,0.14,'PLEASE MOUNT BLACK LIQ-INK PENS'
     1BROAD IN POSITION 1 AND EXTRA FINE IN POSITION 2',90.0,80)
C
      CALL PLOT(0.0,0.0,-3)
      CALL NEWPEN(1)
      CALL PLOT(AAX,AAY,-3)
C
      DO 40 I=1,IFLAG
      N1=1+(I-1)*NS
      N2=NS+(I-1)*NA
      X1=FLOAT(IBUF1(N2))*SX
      Y1=FLOAT(IBUF2(N2))*SY
      CALL PLOT(X1,Y1,3)
      DO 40 N=N1,N2
      X1=FLOAT(IBUF1(N))*SX
      Y1=FLOAT(IBUF2(N))*SY
      CALL PLOT(X1,Y1,2)
   40 CONTINUE
      CALL PLOT(0.0,0.0,3)
      IF(ISYM.NE.1) GO TO 45
      CALL PLOT(10.0,0.0,2)
      CALL PLOT(0.0,0.0,3)
      CALL PLOT(0.0,10.0,2)
      CALL PLOT(0.0,0.0,3)
   45 CALL NEWPEN(2)
C
   47 DO 50 I=1,IEND
      II=IEND+1-I
      NI=NNI(II)
      NJ=NNJ(II)
      IF(II.EQ.1) GO TO 70
      NN=2
      IF(ISYM.EQ.1) NN=1
      IM1=NI*2
      JM1=NJ*2
      IMAX=IM1+1
      JMAX=JM1+1
      GO TO 80
   70 IMAX=NI+1
      JMAX=NJ+1
   80 DO 90 L=1,IMAX
      DO 90 M=1,JMAX
      ZP(L,M)=ZM
   90 CONTINUE
      IP=4
      IF(ISYM.EQ.1) IP=3
      IF(II.EQ.1) GO TO 280
      IF(II.EQ.NESMAX) GO TO 100
      K1=NCONN(II+1,IP,2)+1
      GO TO 110
  100 K1=1
  110 K2=NCONN(II,IP,2)
      I1=IG(K1,1)
      I2=IG(K1,2)
      I4=IG(K1,4)
      X0=XG(I1)
      Y0=YG(I1)
      DDX=(XG(I2)-X0)/2.0
      DDY=(YG(I4)-Y0)/2.0
      IF(ISYM.NE.1) GO TO 115
      X0=0.0
      Y0=0.0
C
  115 DO 120 K=K1,K2
      IF(ISYM.EQ.1) GO TO 124
      J1=1
      J2=4
      GO TO 128
  124 J1=2
      J2=3
  128 DO 130 J=J1,J2
      JJ=J
      IF(K.GE.NCONN(II,J,1).AND.K.LE.NCONN(II,J,2)) GO TO 140
  130 CONTINUE
      JJ=5
      J1=4
      GO TO 150
  140 JJ=J
  150 DO 160 IA=1,J1
      IF(IA.EQ.5) GO TO 170
      N=IG(K,IA)
      GO TO 180
  170 IF(II.GT.2) GO TO 179
      N=0
      J2=JJ-1
      IF(J2.LT.J1) GO TO 174
      DO 172 J=J1,J2
      N=N+NCONN(II,J,2)-NCONN(II,J,1)+1
  172 CONTINUE
  174 J1=NCONN(II,JJ,1)
      J2=NCONN(II,JJ,2)
      DO 176 J=J1,J2
      N=N+1
      IF(K.EQ.J) GO TO 178
  176 CONTINUE
  178 N=NCONA(N)
      GO TO 180
  179 N=(K-NCONN(II,JJ,1))*2+NCONN(II,JJ,3)
  180 L=IFIX((XG(N)-X0)/DDX+0.1)+1
      M=IFIX((YG(N)-Y0)/DDY+0.1)+1
      ZP(L,M)=POT(N)
  160 CONTINUE
      IF(JJ.EQ.5) GO TO 120
      GO TO (190,200,210,220),JJ
  190 L1=0
      M1=-2
      GO TO 230
  200 L1=2
      M1=0
      GO TO 230
  210 L1=0
      M1=2
      GO TO 230
  220 L1=-2
      M1=0
  230 N1=JJ
      N2=JJ+1-JJ/4*4
      N1=IG(K,N1)
      N2=IG(K,N2)
      L2=L+L1
      M2=M+M1
      ZP(L2,M2)=(POT(N1)+POT(N2))/2.0
      L3=L+L1/2
      M3=M+M1/2
      ZP(L3,M3)=(ZP(L2,M2)+POT(N))/2.0
  120 CONTINUE
      DO 240 L=1,IMAX,2
      DO 240 M=2,JM1,2
      Z1=ZP(L,M)
      Z2=ZP(L,M+1)
      Z3=ZP(L,M-1)
      IF(Z1.NE.ZM) GO TO 240
      IF(Z2.EQ.ZM.OR.Z3.EQ.ZM) GO TO 240
      ZP(L,M)=(Z2+Z3)/2.0
  240 CONTINUE
      DO 250 M=1,JMAX
      DO 250 L=2,IM1,2
      Z1=ZP(L,M)
      Z2=ZP(L+1,M)
      Z3=ZP(L-1,M)
      IF(Z1.NE.ZM) GO TO 250
      IF(Z2.EQ.ZM.OR.Z3.EQ.ZM) GO TO 250
      ZP(L,M)=(Z2+Z3)/2.0
  250 CONTINUE
      DO 260 L=1,IMAX
      XP(L)=X0+DDX*FLOAT(L-1)
  260 CONTINUE
      DO 270 L=1,JMAX
      YP(L)=Y0+DDY*FLOAT(L-1)
  270 CONTINUE
      GO TO 320
C
C
  280 DO 290 L=1,IMAX
      N=IL(L,JMAX)
      XP(L)=XG(N)
  290 CONTINUE
      DO 300 L=1,JMAX
      N=IL(IMAX,L)
      YP(L)=YG(N)
  300 CONTINUE
      DO 310 L=1,IMAX
      DO 310 M=1,JMAX
      N=IL(L,M)
      IF(N.EQ.-1) GO TO 310
      IF(MAX.EQ.0) GO TO 306
      DO 305 LL=1,MAX
      L1=(LL-1)*2+1
      IF(N.EQ.IPLOT(L1)) GO TO 310
  305 CONTINUE
  306 ZP(L,M)=POT(N)
  310 CONTINUE
C
C
  320 IF(SC.EQ.1.0) GO TO 345
      DO 330 L=1,IMAX
      XP(L)=XP(L)*SC
  330 CONTINUE
      DO 340 L=1,JMAX
      YP(L)=YP(L)*SC
  340 CONTINUE
  345 IF(SCZ.EQ.1.0) GO TO 355
      DO 350 L=1,IMAX
      DO 350 M=1,JMAX
      ZP(L,M)=ZP(L,M)*SCZ
  350 CONTINUE
  355 IF(MM.EQ.1) GO TO 370
      DO 360 L=1,NCN
      CN=CNP(L)
      CALL CNTOUR(XP,IMAX,YP,JMAX,ZP,IDIMX,CN,SEP,CN)
  360 CONTINUE
  370 IF(MMM.EQ.1) GO TO 475
      WRITE(6,380) (XP(L),L=1,IMAX)
      WRITE(6,380) (YP(L),L=1,JMAX)
  380 FORMAT(13F8.3)
      DO 390 M=1,JMAX
      MASA=JMAX+1-M
      WRITE(6,380) (ZP(L,MASA),L=1,IMAX)
  390 CONTINUE
C
  395 M1=0
      K1=KELOUT+1
      DO 400 K=K1,KMAX
      IF(IG(K,4).NE.0) GO TO 400
      M1=M1+1
      IBUFF(M1)=K
  400 CONTINUE
      IF(M1.EQ.0) GO TO 475
C
      DO 410 M=1,M1
```

```
      K=IPLOT(M)
      DO 420 N=1,4
      IG(N)=IG(K,N)
  420 CONTINUE
C
      DO 430 L=1,4
      K=IG(L)
      XP(L)=XG(K)*SC
      YP(L)=YG(K)*SC
      ZP(1,L)=POT(K)*SCZ
  430 CONTINUE
C
      DO 440 L=1,NCN
      CN=CNP(L)
      K=0
      DO 450 N=1,3
      N1=N+1-N/3*3
      C1=CN-ZP(1,N)
      C2=CN-ZP(1,N1)
      IF(C1*C2.GT.0.0) GO TO 450
      C3=C1-C2
      IF(C3.EQ.0.0) GO TO 460
      C4=C1/C3
      GO TO 470
  460 C4=0.0
  470 K=K+1
      XPP(K)=C4*(XP(N1)-XP(N))+XP(N)
      YPP(K)=C4*(YP(N1)-YP(N))+YP(N)
  450 CONTINUE
C
      IF(K.LE.1) GO TO 440
      CALL PLOT(XPP(1),YPP(1),3)
      CALL PLOT(XPP(2),YPP(2),2)
  440 CONTINUE
  410 CONTINUE
C
  475 IF(MAX.EQ.0) GO TO 50
      DO 480 M=1,MAX
      M1=IPLOT(M82)
      M2=IG(M1,1)
      ZP(1,1)=POT(M2)*SCZ
      M2=IG(M1,4)
      ZP(1,2)=POT(M2)*SCZ
      M2=IG(M1,2)
      ZP(2,1)=POT(M2)*SCZ
      M2=IG(M1,3)
      ZP(2,2)=POT(M2)*SCZ
      XP(1)=XG(IG(M1,1))*SC
      XP(2)=XG(IG(M1,2))*SC
      YP(1)=YG(IG(M1,1))*SC
      YP(2)=YG(IG(M1,4))*SC
      DO 490 L=1,NCN
      CN=CNP(L)
      CALL CNTOUR(XP,2,YP,2,ZP,IDIMX,CN,SEP,CN)
  490 CONTINUE
  480 CONTINUE
C
   50 CONTINUE
      IF(MM.EQ.1) GO TO 500
      CALL FINISH
  500 STOP
      END
      SUBROUTINE PAPLT(NCN,CNP)
      DIMENSION CNP(10)
      DATA NCN/2/
      DATA CNP/50.0,55.0/
      RETURN
      END
      END$
```

FILM